

Mikrokomputery

Donald Hearn M. Pauline Baker

GRAFIKA MIKRO-KOMPUTEROWA

GRA FIKA

Grafika mikrokomputerowa

Mikrokomputery

Donald Hearn
M. Pauline Baker

Grafika mikrokomputerowa

metody i zastosowania

Z języka angielskiego przełożyli
MARIAN ŁAKOMY
JAN ZABRODZKI



Wydawnictwa Naukowo-Techniczne
Warszawa 1988

Dane o oryginale

MICROCOMPUTER GRAPHICS
TECHNIQUES AND APPLICATIONS
Donald Hearn
M. Pauline Baker

Computer Science Division
Western Illinois University

Original English language edition published
by Prentice-Hall, Inc., Englewood Cliffs,
New Jersey, USA

Redaktor *Lilianna Blaszkiewicz, Zuzanna Grzejszczak*
Redaktor techniczny *Irena Milewska-Burczykowa*
Projektant serii *Juliusz Rybicki*
Okładkę i strony tytułowe projektował *Andrzej Pilich*

681.32

Książka zawiera zbiór podstawowych wiadomości z grafiki komputerowej. Przedstawiono różne metody rysowania, transformacji i animacji obrazów. Omówiono istotę grafiki trójwymiarowej. Prezentację różnorodnych zastosowań grafiki w nauczaniu, domu, handlu itp. wzbogacono licznymi przykładami programów. Praca zawiera wiele rozwiązań praktycznych.
Książka jest przeznaczona dla szerokiego kręgu czytelników.

ISBN 83-204-0988-8

Spis treści

Część II
PODSTAWY GRAFIKI/35
(Początki)

Część III
ŚREDNIO ZAAWANSOWANE TECHNIKI GRAFICZNE/107
(Co więcej możemy zrobić?)

Lista przykładowych programów

Wykaz barwnych fotografii

Fot. A. Trójwymiarowe obrazy mogą być stosowane do wstępnej oceny projektu. (Za zgodą Lexidata Corp.)

Fot. B. Metody CAD są używane przy projektowaniu elementów elektronicznych, takich jak na przykład pokazany obwód drukowany. (Za zgodą Lexidata Corp.)

Fot. C. Za pomocą komputerowych systemów graficznych artyści projektują rysunki i filmy animowane. (Za zgodą Chromatics Inc.)

Fot. D. Na ekranie monitora graficznego można tworzyć rozmaite rysunki abstrakcyjne. (Za zgodą Los Alamos National Laboratory)

Fot. E. Z grafiki komputerowej korzystają artyści przy projektowaniu orientalnych deseni dywanów. (Za zgodą Lexidata Corp.)

Fot. F. Zestaw komputerowy dla artystów projektujących reklamy przy użyciu metod przetwarzania obrazów. (Za zgodą Lexidata Corp.)

Fot. G. W zastosowaniach medycznych korzysta się z metody przetwarzania obrazów do obserwacji działania systemów fizjologicznych. (Za zgodą Lexidata Corp.)

Fot. H. Wykres komputerowy 16 milionów punktów zaobserwowanych przez astronomów w mgławicy Whirlpool dzięki zakodowaniu względnych jasności za pomocą kolorów, ujawnił duże galaktyki. (Za zgodą Los Alamos National Laboratory)

Fot. I. Model komputerowy powierzchni wykorzystywanej w badaniach zdarzeń atomowych i jądrowych. (Za zgodą Los Alamos National Laboratory)

Fot. J. Trójwymiarowy wykres ciśnienia. (Reprodukcja za zgodą ISSCO, San Diego, CA)

Fot. K. Trójwymiarowy wykres może ilustrować różne zależności na jednym diagramie. (Reprodukcja za zgodą ISSCO, San Diego, CA)

Fot. L. Kolorowy diagram ilustrujący działanie reaktora jądrowego. (Za zgodą Los Alamos National Laboratory)

Fot. M. Widok pasa do lądowania wygenerowany za pomocą komputera dla potrzeb symulatora lotu przeznaczonego do trenowania pilotów. (Za zgodą Evans and Sutherland)

Fot. N. Łączenie wykresów jest efektywnym sposobem przekazywania różnego rodzaju informacji za pomocą jednego obrazu. (Reprodukcja za zgodą ISSCO, San Diego, CA)

Fot. O i P. Możliwe są różne konstrukcje wykresów trójwymiarowych. (Reprodukcja za zgodą ISSCO, San Diego, CA)

Przedmowa

Książka stanowi wprowadzenie do grafiki komputerowej, przy czym specjalny nacisk położono na metody stosowane w mikrokomputerach. Wobec dostępności tanich mikrokomputerów grafika może być powszechnie stosowana. W książce pokazujemy możliwości małych systemów komputerowych w różnych zastosowaniach, włączając w to animację i generację dwu- i trójwymiarowych obrazów i wykresów.

Książkę podzieliliśmy na pięć części. W części I podajemy przegląd sposobów korzystania z grafiki komputerowej w projektowaniu, przetwarzaniu obrazów, handlu, sztuce, nauczaniu, badaniach i w domu. Omówiono ogólne cechy sprzętu i oprogramowania systemów graficznych; podano również przegląd możliwości wybranych mikrokomputerów.

W części II po omówieniu podstaw grafiki wprowadzamy metody tworzenia obrazów za pomocą znaków i pikseli. Podano metody rysowania obrazów i kreślenia wykresów, łącznie z cieniowaniem i kolorami.

Efektom specjalnym poświęcamy część III. Omawiamy transformacje obrazu, animację, metody wyróżniania i obcinania elementów obrazu.

W części IV przedstawiamy grafikę trójwymiarową. Analizujemy metody usuwania niewidocznych linii, tworzenia rzutów perspektywicznych i transformacji trójwymiarowych obrazów. Dyskutujemy zastosowanie tych metod zarówno w odniesieniu do obrazów, jak i wykresów.

Użycie metod grafiki komputerowej w zastosowaniach handlowych, w nauczaniu i w domu rozważamy w części V. W tej końcowej części zajmujemy się metodami tworzenia programów, specjalnymi metodami grafiki, symulacją, nauczaniem za pomocą komputera, wykresami budżetu domowego i żywienia oraz grami.

Metody grafiki omawiane w książce zilustrowano programami napisanymi w Basicu, standardowym języku stosowanym w większości mikrokomputerów. Wszystkie programy opracowano i sprawdzono w systemach mikrokomputerowych. Podane przykłady mają na celu wyjaśnienie szczegółów realizacji programów graficznych. Wiele rozkazów w tych przykładach można by zmienić lub pominąć przy

pisaniu bardziej efektywnych programów; zostały one jednak świadomie podane w takiej postaci po to, by można było w łatwy sposób wyjaśnić kolejne kroki postępowania. W większości przypadków długość nazw zmiennych w programach ograniczono do dwóch znaków – w niektórych mikrokomputerach występuje takie ograniczenie. Dłuższych nazw zmiennych użyliśmy w tych nielicznych sytuacjach, w których zależało nam na jasności dokumentacji programu. Ponieważ nie istnieje jednolity zestaw rozkazów graficznych obowiązujący dla wszystkich systemów, w naszych przykładowych programach korzystamy z hybrydowego zbioru rozkazów graficznych adaptowanych spośród rozkazów dostępnych w różnych mikrokomputerach. Powiązanie tego zbioru hipotetycznych rozkazów graficznych z rzeczywistymi rozkazami dla określonych mikrokomputerów podano w dodatku A.

DONALD HEARN
M. PAULINE BAKER

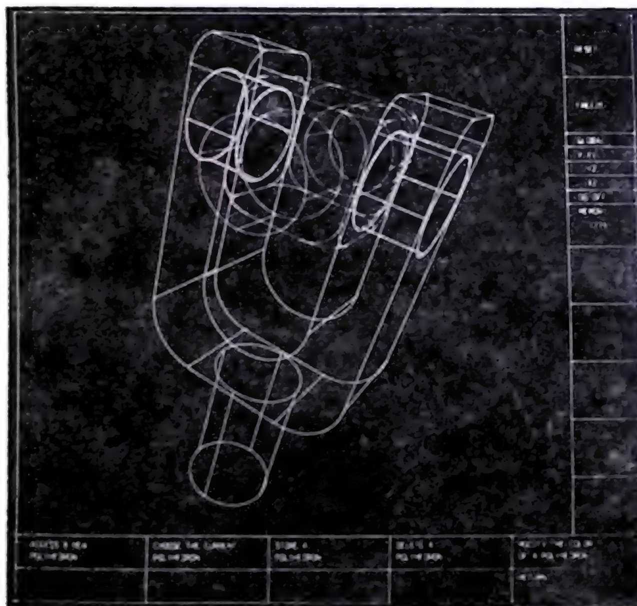
Część I
WSTĘP
(Czym jest grafika mikrokomputerowa?)

Od dawna znano znaczenie *obrazu* jako środka szybkiego i dokładnego przekazywania informacji. Powstały różne metody tworzenia obrazów: druk, fotografia, reprodukcja. Obecnie nowoczesny komputer wprowadza nas w erę *grafiki komputerowej*. W celu zapoznania z tą dziedziną omówimy niektóre z wielu możliwych zastosowań. Zobaczymy, jak różni ludzie korzystają z grafiki komputerowej oraz jakich urządzeń i metod używają. Następnie pokażemy możliwości mikrokomputerów w zastosowaniach graficznych.

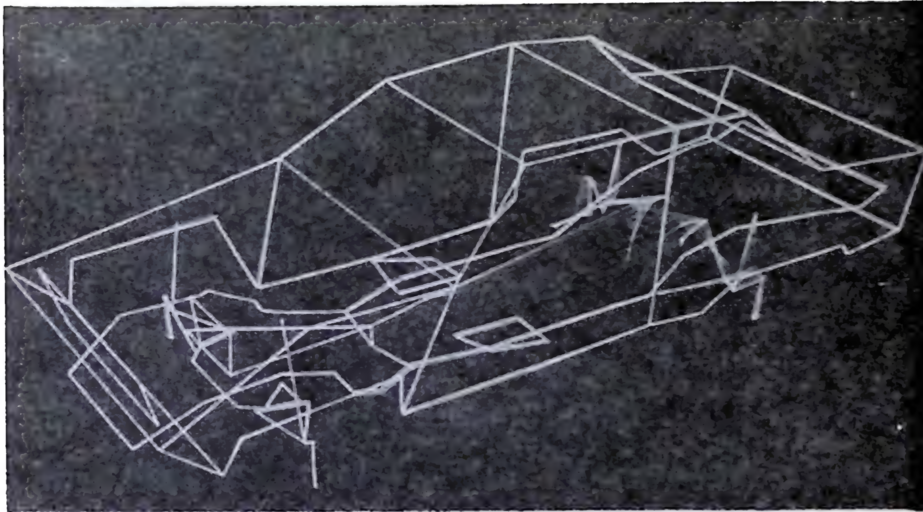
1. Prezentacja możliwości grafiki komputerowej

1.1. Kto korzysta z grafiki komputerowej?

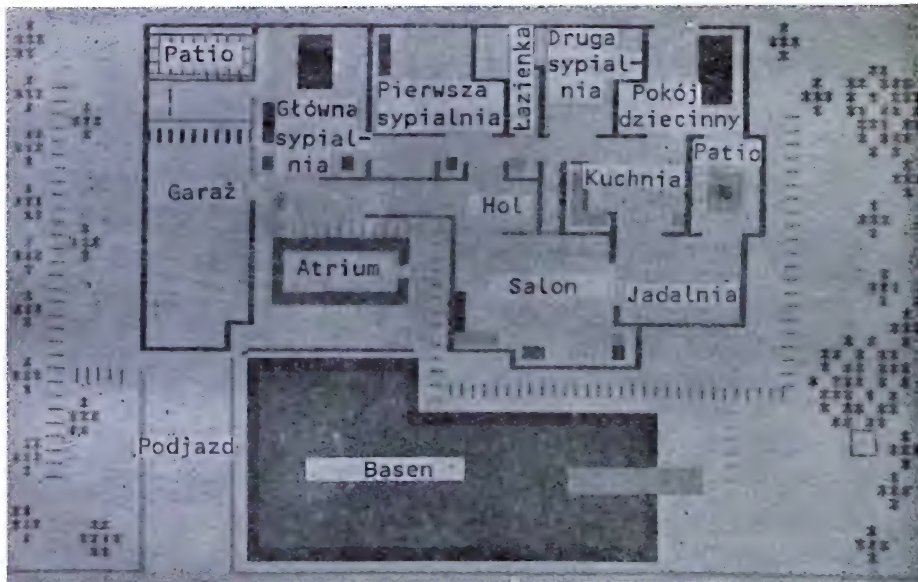
Jedno z pierwszych zastosowań grafiki komputerowej było związane ze wspomaganiem procesu projektowania. *Projektowanie wspomagane komputerem* (CAD) i *wytwarzanie wspomagane komputerem* (CAM) nadal pozostają głównymi obszarami zastosowań grafiki. Ekran komputera umożliwia automatyczne kreślenie rysunków



Rys. 1.1. Popularnym zastosowaniem metod komputerowo-wspomaganego projektowania (CAD) jest kreślenie rysunków złożeniowych (za zgodą Evans and Sutherland)



Rys. 1.2. W początkowej fazie projektowania bryły samochodu, samolotu lub statku kosmicznego można korzystać ze szkiców szkieletowych wytwarzanych za pomocą metod CAD (za zgodą Evans and Sutherland)

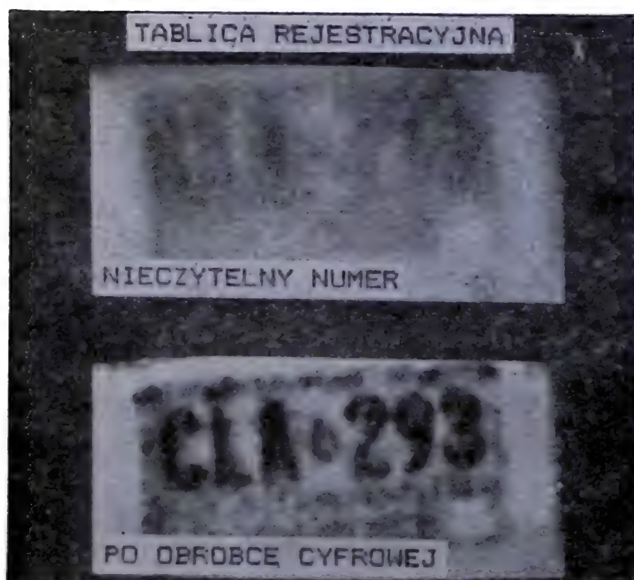


Rys. 1.3. W projektowaniu architektonicznym są użyteczne plany budynków (za zgodą Intelligent Systems Corp.)

technicznych, planów architektonicznych, ogłoszeń i reklam oraz nadzór nad procesami produkcyjnymi. Systemy CAD umożliwiają oglądanie części maszyn pod różnymi kątami. Korzystając z metod grafiki można rysować i wyświetlać rysunki potrzebne przy produkcji danego elementu. Z rysunków tych można korzystać przy określeniu trajektorii narzędzi nad powierzchniami elementu w czasie jego wytwarzania. Umożliwia to z kolei programowanie maszyn sterowanych numerycznie zgodnie z potrzebami procesu wytwarzania części według przygotowanych rysunków.

Projektanci samochodów, samolotów i stacji kosmicznych korzystają z metod CAD przy projektowaniu konturów powierzchni. Rysunki szkieletowe mogą być wyświetlane na ekranie w celu sprawdzenia kształtów samochodów, samolotów czy statków kosmicznych. Dzięki tym rysunkom można obejrzeć kontur całej powierzchni lub pojedynczych części, jak błotnik samochodu czy skrzydła samolotu. W każdym kroku procesu projektowania można dodawać coraz więcej detali do komputerowo generowanego obrazu. Końcowy, realistyczny obraz obiektu umożliwia projektantowi obejrzenie projektowanego wyrobu.

Metodami CAD są projektowane układy elektryczne i elektroniczne. Korzystając z *piktogramów* reprezentujących różne elementy, inżynier elektronik może tworzyć układ na ekranie. Dzięki obrazowi planu budynku inżynier elektryk może wypróbować różne warianty rozmieszczenia elektrycznych gniazdek lub systemów sygnalizacji pożarowej.



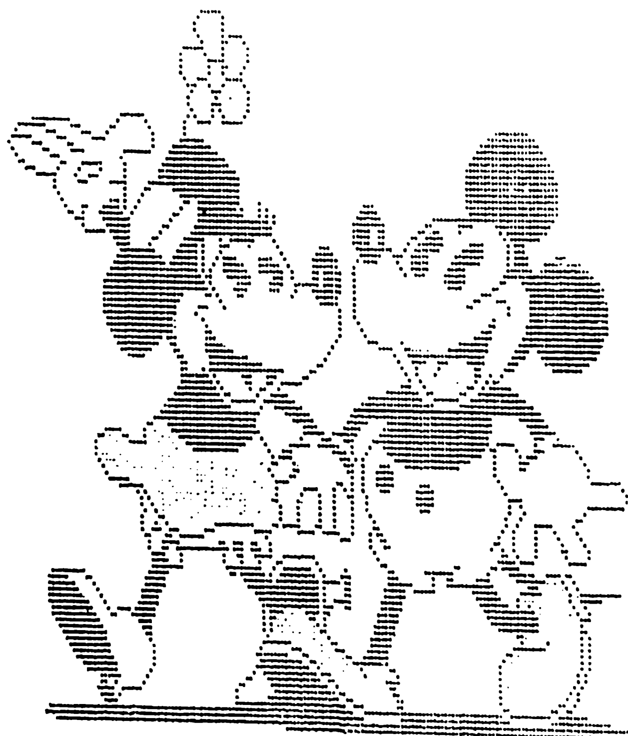
Rys. 1.4. Metody przetwarzania obrazów umożliwiają odczytanie zamazanych numerów rejestracyjnych (za zgodą Los Alamos National Laboratory)



Rys. 1.5. Mapa pogody uzyskana metodami grafiki komputerowej (za zgodą Genisco Computers Corp.)



Rys. 1.6. Finansowy wykres słupkowy (za zgodą Precision Visuals, Inc., Boulder, CO)



Rys. 1.7. Drukarka umożliwia uzyskanie prostych rysunków do różnych zastosowań domowych

Również architekci korzystają w procesie projektowania z planów budynków wytwarzanych metodami CAD. Plany te, wyświetlane w różnych postaciach, są użyteczne przy projektowaniu wyposażenia pokoju, rozmieszczania drzwi i okien i innych urządzeń. Trójwymiarowe obrazy umożliwiają oglądanie projektu jednego budynku lub sprawdzanie wyglądu złożonego kompleksu budynków (alei, ośrodka uniwersyteckiego, kompleksu szpitalnego).

Z metod grafiki komputerowej korzystają artyści. Za pomocą monitorów graficznych wytwarza się filmy animowane. Systemy graficzne generują abstrakcyjne i geometryczne wzory dla projektantów mody. Wzory generowane komputerowo występują w wielu zastosowaniach komercyjnych, włączając w to projektowanie tkanin. Artyści korzystają również z metod przetwarzania obrazów w zastosowaniach takich, jak retuszowanie czy poprawianie zdjęć lub innych obrazów.

Technika przetwarzania obrazów umożliwia wyświetlanie na ekranie obrazów uzyskanych z kamery telewizyjnej. Chociaż technika ta stosuje komputery do generowania obrazów, różni się ona od konwencjonalnych metod grafiki komputerowej. W grafice komputerowej obraz jest „tworzony" na ekranie przez system

graficzny dla potrzeb danego zastosowania. W przetwarzaniu obrazów dokonuje się *digitalizacji* zdjęcia lub obrazu z kamery i uzyskaną informację przesyła na ekran. Stosuje się przy tym metody umożliwiające wyróżnianie fragmentów obrazu i wzmacnianie różnic kolorów lub cieni dla poprawienia jakości obrazu. W medycynie korzysta się z metod przetwarzania obrazów uzyskiwanych za pomocą promieni X do obserwowania działania organów wewnętrznych. Te same metody są użyteczne przy oglądaniu dowolnych systemów lub obiektów, których nie można obserwować bezpośrednio, na przykład obrazów telewizyjnych przesyłanych ze stacji orbitalnych lub obrazów z „oka" robota przemysłowego.

W wielu dziedzinach naukowcy traktują grafikę komputerową jako ważne narzędzie w badaniu charakterystyk systemów. Astronomowie, zbierając dane o gwiazdach i galaktykach, opracowują modele graficzne, które pomagają wyjaśnić budowę i zachowanie obiektów kosmicznych. Bez pomocy takich wykresów tablice danych zawierające miliony wartości byłyby trudne do interpretacji. Również w celu lepszego zrozumienia budowy systemów biologicznych, fizycznych i chemicznych tworzy się odpowiednie modele graficzne. Przy interpretacji zależności matematycznych lub badaniach trendów w zachowaniu systemów obok modeli korzysta się z wykresów generowanych przez komputery.

Dane zamieszczane na wykresach mogą przyjmować różne postacie. Na przykład na podstawie danych dostarczanych przez stacje obserwacyjne systemy graficzne mogą konstruować dwuwymiarowe mapy pogody. Na trójwymiarowych wykresach można porównywać zmiany ciśnienia lub temperatury nad obszarami geograficznymi.

Z podobnych metod korzysta się w wielu zastosowaniach komercyjnych i administracyjnych. Ten obszar zastosowań interesuje największą liczbę użytkowników grafiki komputerowej. Do reprezentacji danych finansowych i statystycznych używa się różnego rodzaju wykresów liniowych i słupkowych. Pokazanie złożonych zależności odbywa się za pomocą obrazów i wykresów trójwymiarowych. Do wizualizacji różnego typu statystyk dotyczących regionów korzysta się z wykresów typu „mapa kraju". Tego rodzaju obrazy są często włączane do sprawozdań dla dyrekcji lub do biuletynów informacyjnych dla konsumentów, lub też jako pomoc wizualna do wykładów.

Grafika komputerowa jest stosowana do celów dydaktycznych i szkoleniowych. Obrazy i wykresy służą do wyjaśnienia działania różnych systemów. Przy szkoleniu pilotów i dowódców okrętów symuluje się rzeczywiste wrażenia wzrokowe. Z obrazów graficznych korzysta się w pokazach w czasie lekcji, egzaminów sterowanych komputerem, w programach studiów zaocznych.

Metody grafiki są przydatne również do osobistych potrzeb. Drukowane obrazy mogą być używane w prywatnej korespondencji, np. w postaci własnoręcznie projektowanych kart okolicznościowych, do dekoracji ścian. Obrazy wyświetlane na ekranie są wykorzystywane w grach lub w programach edukacyjnych, a wykresy generowane przez komputer przy kontroli diety, budżetu czy też lokaty kapitału.

1.2. Projektowanie i wyświetlanie obrazów komputerowych

Zależnie od zastosowania korzysta się z różnych systemów grafiki komputerowej. Systemy *wysoko specjalizowane* buduje się dla potrzeb symulatorów lotu przeznaczonych do trenowania pilotów. Systemy specjalizowane spełniają określone zadania, takie jak wyświetlanie widoków lotnisk czy też pokładów lotniskowców. Możliwość tworzenia różnego rodzaju obrazów i wykresów zapewniają systemy *uniwersalne*.

Systemy grafiki komputerowej

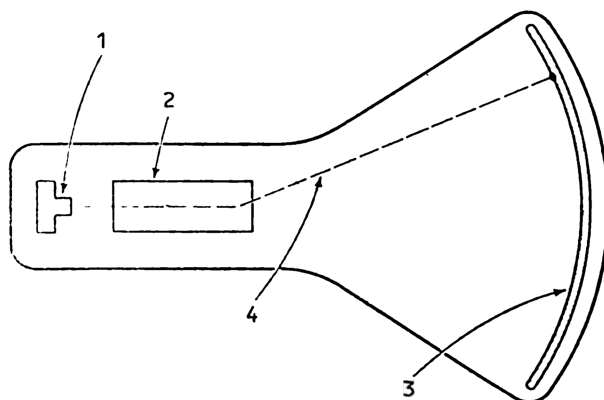
Typowymi elementami systemu *uniwersalnego* grafiki komputerowej są: urządzenie wyświetlające z ekranem, klawiatura, programy graficzne i oczywiście jednostki procesora i pamięci komputera. Do większości systemów graficznych można przyłączyć dodatkowe urządzenia wejściowe i wyjściowe inne niż klawiatura i monitor.

Możliwa jest współpraca monitora z *tabliczką graficzną* (ang. *graphics tablet*), która służy do wprowadzania (*digitalizacji*) rysunku czy projektu wykonanego na papierze na ekran lub do pamięci komputera. Można potem reprodukować zapamiętany (np. na dysku) obraz za pomocą plotera. Obrazy z ekranu są często reprodukowane na przeźroczach 35 mm lub na foliach do wyświetlania przez rzutniki pisma. Łącząc kilka obrazów na taśmie filmowej można uzyskać sekwencję obrazów ruchomych. Tabliczka graficzna jest zmniejszoną wersją *digitizera* (ang. *digitizing board*) i może być używana do tych samych zadań. Rysunki zapamiętane w postaci cyfrowej mogą być wyświetlane na monitorze, a następnie modyfikowane lub łączone z innymi rysunkami w celu tworzenia bardziej złożonych rysunków.

Monitory

Większość urządzeń wyświetlających stosowanych w grafice komputerowej jest wyposażona w *lampę obrazową* (CRT). Zasadę działania lampy CRT zilustrowano na rys. 1.8. Strumień elektronów (promienie katodowe), emitowany przez wyrzutnię elektronową, przechodzi przez układy ogniskowania i odchylania i uderza w ekran pokryty luminoforem. Liczba emitowanych elektronów zależy od wartości napięć zasilających. Układ ogniskowania i odchylania jest sterowany również *napięciowo*. Układ ten wytwarza pola elektryczne i magnetyczne skupiające strumień w określonym punkcie ekranu. Skutkiem uderzenia strumienia elektronów w luminofor jest rozświetlenie ekranu w tym miejscu. Intensywność świecenia punktu zależy od liczby elektronów w strumieniu. Kierując strumień elektronów w różne punkty ekranu możemy wyświetlić obraz.

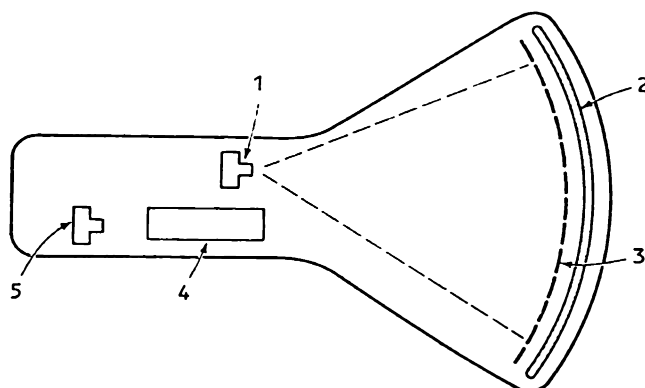
Emisja światła przez luminofor pokrywający ekran trwa tylko przez ułamek sekundy. Stąd potrzebna jest jakaś metoda utrzymania obrazu na ekranie tak, by można było ten obraz oglądać. Są dwa podstawowe rozwiązania tego problemu.



Rys. 1.8. Zasada działania lampy obrazowej
1 — wyrzutnia elektronowa; *2* — układ ogniskowania i odchylania; *3* — ekran pokryty luminoforem; *4* — strumień elektronów

Pierwsza metoda podtrzymania świecenia luminoforu na ekranie monitora polega na wielokrotnym kierowaniu strumienia elektronów w te same punkty ekranu. Ten typ wyświetlania jest określany jako *odnawianie obrazu* na ekranie lampy. Okazuje się, że aby uniknąć efektu migotania, konieczne jest odnawianie obrazu na ekranie ok. 30 razy na sekundę. Typowe monitory *z odnawianiem* umożliwiają rysowanie obrazu od 30 do 60 razy na sekundę.

Druga metoda utrzymania obrazu na ekranie polega na stosowaniu dodatkowej wyrzutni podtrzymującej i specjalnego ekranu (siatki kolektorowej), który pamięta obraz w postaci rozkładu ładunku elektrycznego (rys. 1.9). Takie rozwiąza-



Rys. 1.9. Zasada działania lampy pamięciowej
1 — wyrzutnia podtrzymująca; *2* — luminofor; *3* — siatka kolektorowa; *4* — układ ogniskowania i odchylania; *5* — wyrzutnia elektronowa

nie jest stosowane w lampach *pamięciowych* (DVST). Strumień elektronów przebiega cały ekran jeden raz i w trakcie tego procesu na *siatce kolektorowej* jest zapamiętywany wzór ładunków, odpowiadający wyświetlonemu obrazowi. *Wyrzutnia podtrzymująca* emituje ciągły strumień elektronów o niewielkiej prędkości. Elektrony te aktywują luminofor zgodnie ze wzorem ładunków na siatce kolektorowej. Umożliwia to utrzymanie świecenia luminoforu w odpowiednich miejscach ekranu.

Monitory projektowane z wykorzystaniem lamp obrazowych wymagających odnawiania obrazu mogą działać na zasadzie przeszukiwania *przypadkowego* (monitory wektorowe) bądź przeszukiwania *sekwencyjnego* (monitory rastrowe). W monitorach *wektorowych* strumień elektronów jest kierowany tylko na te części ekranu, gdzie ma być narysowany obraz. W monitorach *rastrowych* strumień elektronów przebiega cały ekran, przy czym intensywność strumienia jest zmieniana zależnie od struktury obrazu. Przykładem urządzenia z przeszukiwaniem sekwencyjnym jest telewizor. W systemie rastrowym strumień elektronów przebiega wzdłuż kolejnych linii poziomych ekranu od góry do dołu. Często cykl odnawiania jest tak organizowany, aby strumień elektronów, podczas jednego przejścia od góry do dołu ekranu, przebiegał wzdłuż linii parzystych, a przy kolejnym przejściu wzdłuż linii nieparzystych. Takie przeplecenie przeszukiwanych linii (zwane *wybieraniem międzyliniowym*) zmniejsza migotanie. Wynika to z faktu, że w zasadzie widzimy obraz całego ekranu w czasie równym połowie czasu potrzebnego na przeszukanie wszystkich linii od góry do dołu.

W celu uzyskania obrazu kolorowego na ekranie używa się więcej niż jednego rodzaju luminoforu pokrywającego ekran. Różne luminofory emitują światło o różnych barwach, a kombinacja barw podstawowych pochodzących z dwóch lub więcej luminoforów może dać pełną gamę kolorów.

W wielu monitorach *kolorowych* stosowanych w grafice komputerowej używa się lampy obrazowej z maskownicą. Ten typ lampy obrazowej jest stosowany w odbiornikach telewizji kolorowej. W lampie z *maskownicą* ekran jest pokryty trójkami „pastylek" (elementarnych powierzchni) luminoforów ułożonych w wierzchołkach trójkąta. W każdym trójkącie jedna „pastylka" emituje światło czerwone, druga zielone, trzecia zaś niebieskie. Ten typ lampy ma trzy wyrzutnie elektronowe, po jednej na każdy kolor, i maskownicę umieszczoną tuż przed ekranem. Zadaniem maskownicy jest skupienie strumieni elektronów z każdej wyrzutni tak, aby uderzały one tylko we właściwe „pastylki" wybranego trójkąta. Kombinacja poziomów intensywności każdej z wyrzutni elektronowych wyznacza barwę wybranego trójkąta – każdy trójkąt jest tak mały, że odnosi się wrażenie, iż na ekranie obserwuje się jeden kolorowy punkt.

W niektórych monitorach graficznych w celu otrzymania obrazu kolorowego korzysta się z metody penetracji strumienia. W tym przypadku ekran jest pokryty dwiema warstwami luminoforu, zazwyczaj czerwoną i zieloną. Strumień wolniejszych elektronów pobudza tylko czerwoną warstwę. Strumień szybszych elektronów przenika przez luminofor czerwony do warstwy zielonej, wytwarzając światło będące

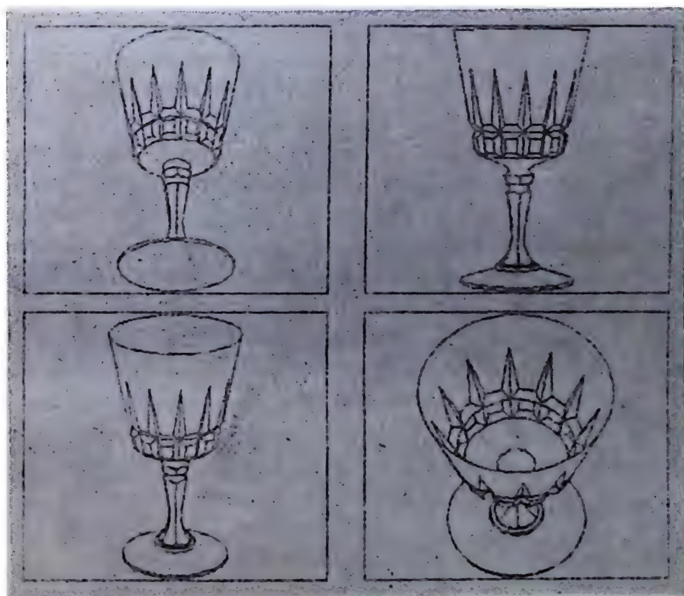
pewną kombinacją czerwieni i zieleni. Prędkość elektronów, a tym samym barwa punktu ekranu, jest określana przez napięcie przyśpieszające elektrony.

Do budowy monitorów korzysta się również z innych technik niż te, w których stosuje się lampy obrazowe. Można wymienić urządzenia z wyświetlaniem plazmowym, z przeszukiwaniem laserowym, z diodami świecącymi (LED), z ciekłymi kryształami (LCD).

Urządzenia wyświetlające do zastosowań graficznych są dostępne zarówno jako niezależne terminale, jak też jako części systemów komputerowych. *Terminale graficzne* mają zazwyczaj niewielką moc obliczeniową (lub w ogóle jej nie mają) i muszą być połączone z zewnętrznym komputerem (może to być niewielki system lub komputer o dużej mocy obliczeniowej). Niektóre monitory graficzne mają wbudowane procesory i mogą pracować samodzielnie bez konieczności łączenia z zewnętrznym komputerem. Terminale i kompletne systemy graficzne są dostępne w różnych wersjach, w zależności od możliwości funkcjonalnych i wymiarów ekranu.

Programowanie dla celów grafiki

Poziomy napięć doprowadzanych do urządzenia wyświetlającego w celu wytworzenia określonych wzorów i barw na ekranie są generowane w wyniku realizacji rozkazów graficznych występujących w *programie wyświetlania*. Programy



Rys. 1.10. Obrót obiektu trójwymiarowego (za zgodą Selanar Corp.)

te są często pisane w językach Fortran, Pascal, Basic lub w języku asemblera; korzysta się przy tym ze specjalnych instrukcji graficznych. Dla niektórych zastosowań opracowano również specjalizowane *języki graficzne*.

Instrukcje graficzne są tłumaczone przez procesor komputera na poziomy napięć sterujące jednostkę wyświetlającą tak, aby wytworzyć na ekranie żądany obraz. Programy graficzne mogą tworzyć obrazy lub wykresy statyczne, zmieniać punkt obserwacji obiektu, wyświetlać sceny animowane.

2. Mikrokomputery i ich możliwości graficzne

Rozwój technologii mikroprocesorów doprowadził do opracowania niewielkich systemów komputerowych o dużych możliwościach graficznych. Takie przenośne systemy są budowane z myślą o zastosowaniach graficznych w biurze, w badaniach, w projektowaniu, w szkołach i jako komputery osobiste. Dzięki niedużej cenie, małym rozmiarom i uniwersalności są to systemy bardzo przydatne do różnorodnych zastosowań graficznych.

Niektóre mikrokomputery mają wbudowany monitor, do innych zaś trzeba przyłączać niezależny terminal lub odbiornik telewizyjny. Wiele systemów wyposażonych w funkcje graficzne może tworzyć obrazy zarówno kolorowe, jak i czarno-białe. Systemy graficzne często charakteryzuje się liczbą dostępnych kolorów i liczbą punktów (tzw. rozdzielczością). które można narysować na ekranie. *Rozdzielczość* systemu graficznego jest określana przez liczby punktów możliwych do wyświetlania w poziomie i w pionie, liczby te zaś zależą od systemu grafiki komputerowej oraz od typu stosowanego urządzenia wyświetlającego. W systemach korzystających ze standardowych monitorów telewizyjnych największe wartości rozdzielczości wynoszą ok. 512 punktów w poziomie i 256 punktów w pionie. Znacznie większe rozdzielczości mają specjalne terminale graficzne i monitory.

Typowymi urządzeniami pamięciowymi używanymi w mikrokomputerach są pamięci kasetowe, dyski elastyczne i dyski sztywne. W wielu systemach dostępne są drukarki, plotery, tablice graficzne i pióra świetlne. Mały system komputerowy może być wykorzystany jako niezależna stacja graficzna lub też może być połączony z większym komputerem. W połączeniu z dużym komputerem mikrokomputery pracują jako inteligentne terminale. Niekiedy tworzy się sieci mikrokomputerowe tak, aby zapewnić, na zasadzie podziału czasu, dostęp do baz danych i do wysokiej jakości drukarek i ploterów.

Językiem programowania wyższego poziomu używanym zwykle w mikrokomputerach jest Basic. Do innych języków stosowanych w niektórych systemach należą: Pascal, Fortran, Cobol, APL, PL/1, Forth i specjalizowane pakiety graficzne.

Możliwości graficzne poszczególnych języków są różne, poczynając od zwykłych rozkazów drukowania znaków, kończąc zaś na instrukcjach rysowania i manipulacji złożonymi obrazami.

W następnych punktach przyjrzymy się kilku charakterystycznym mikrokomputerowym systemom graficznym. Niektóre z tych systemów były projektowane głównie do zastosowań graficznych; jednak w większości są to uniwersalne komputery, które mogą mieć różne zastosowania, nie tylko związane z grafiką.

Komputery Apple

Komputery Apple istnieją zarówno jako systemy 8-bitowe, jak i 16-bitowe. W mikrokomputerach 8-bitowych serii Apple II i Apple III wykorzystuje się mikroprocesor 6502. W projekcie przewidziano, że komputery te będą używane z zewnętrznym monitorem lub odbiornikiem telewizyjnym. W różnych trybach graficznych jest kilka poziomów rozdzielczości. Największa rozdzielczość w Apple II wynosi 280 × 192, w Apple III zaś 560 × 192. Istnieją liczne funkcje graficzne; większość trybów graficznych dopuszcza 16 kolorów.



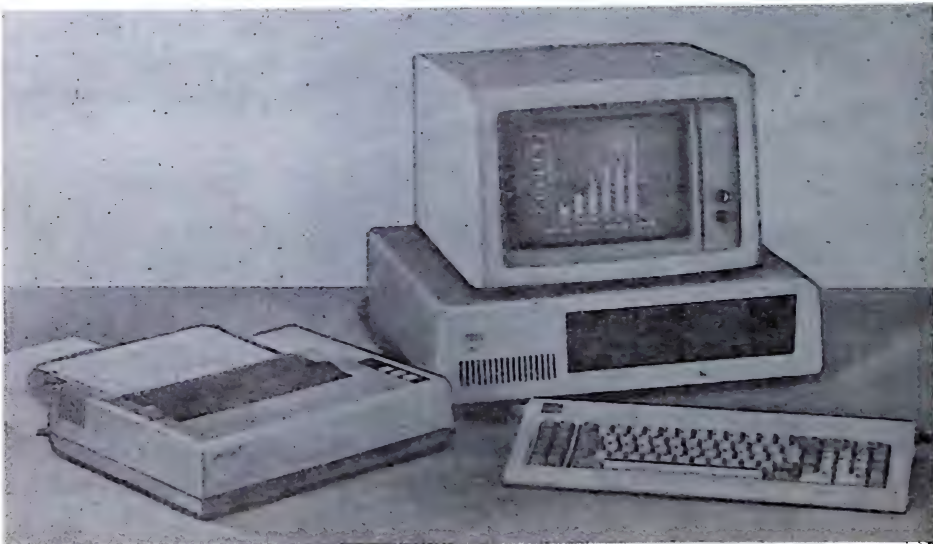
Rys. 2.1. Color Computer firmy Radio Shack z monitorem kolorowym, stacją dysków i drukarką (za zgodą Radio Shack, oddział Tandy Corp.)

Komputery Radio Shack

W komputerach 8-bitowych TRS-80 wykorzystuje się mikroprocesor Z80, a jedynie w „Color Computer" procesor 6809. W systemach 16-bitowych z rodziny TRS-80 jest stosowany mikroprocesor MC68000. Większość systemów firmy Radio Shack ma wbudowane urządzenie wyświetlające. System TRS-80 Color Computer może być używany z odbiornikiem telewizyjnym lub z zewnętrznym monitorem. Jednostka Model II Graphics Option ma kilka rozkazów grafiki czarno-białej o rozdzielczości 640×240. Systemy TRS-80 Color Computer z jednostką Extended Color Basic Option umożliwiają korzystanie z 8 kolorów przy rozdzielczości 256×192.

Komputery osobiste IBM

W tym 16-bitowym komputerze jest zastosowany mikroprocesor Intel 8088. Komputer współpracuje z zewnętrznym monitorem lub z odbiornikiem telewizyjnym



Rys. 2.2. System IBM PC z monitorem monochromatycznym i drukarką (za zgodą IBM Corp.)

Karta grafiki kolorowej umożliwia korzystanie z dwóch rozdzielczości: 320×200 i 640×200. Przy mniejszej rozdzielczości można korzystać z 16 kolorów z dwóch palet (dwóch zestawów kolorów). Przy obu rozdzielczościach można korzystać z licznych funkcji graficznych.

Komputery Atari

Mikrokomputery serii Atari 400 i 800 wykorzystują mikroprocesor 6502. Do jednostek tych trzeba przyłączyć odbiornik telewizyjny lub zewnętrzny monitor. Komputery Atari w trybach graficznych od 3 do 8 mają rozdzielczość z zakresu od 40 × 20 do 320 × 192. Instrukcje graficzne i różne tryby graficzne umożliwiają korzystanie z różnych kombinacji 16 dostępnych kolorów przy określaniu obramowania obrazu oraz jego tylnego i przedniego planu.



Rys. 2.3. Mikrokomputer Atari 800 z monitorem, stacją dysków, pamięcią kasetową i drukarką (za zgodą Atari Inc.)

Komputery Commodore

Firma Commodore Computer Systems dostarcza mikrokomputery wyposażone w klawiatury ze znakami graficznymi. Do systemów tych należą komputery serii PET i CBM: większość z nich ma wbudowany monitor. Mikrokomputery VIC-20 (z mikroprocesorem 6502), Commodore-64 i PET II wymagają przyłączenia



Rys. 2.4. Mikrokomputer Commodore CBM ze stacją dysków i drukarką (za zgodą Commodore Bussiness Machines, Inc.)

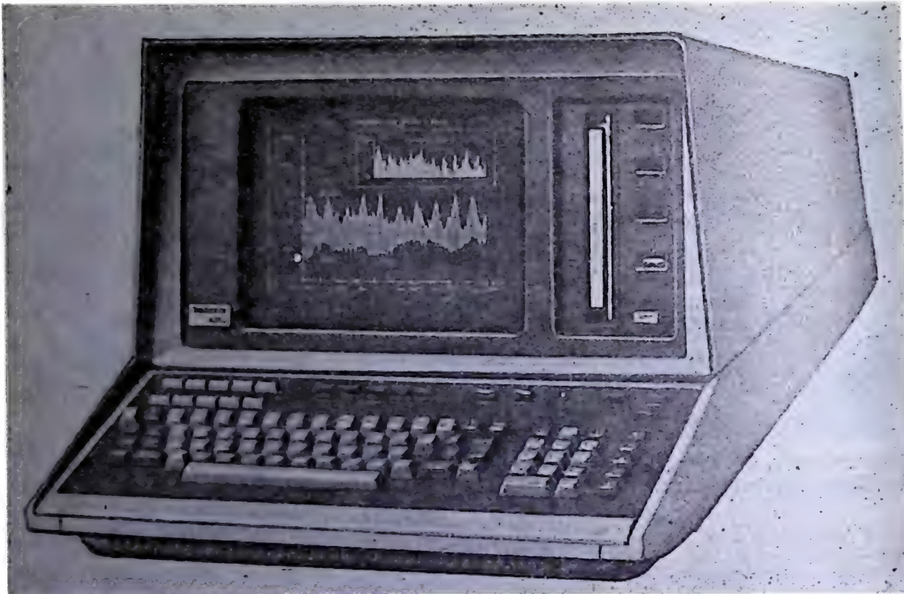
zewnętrznego kolorowego monitora lub odbiornika telewizyjnego. Dodatkowe moduły, np. Super Expander dla VIC-20, zwiększają możliwości graficzne w postaci specjalnych instrukcji graficznych.

Komputery Hewlett-Packard

Firma ta produkuje szereg systemów graficznych, poczynając od komputerów osobistych serii 80 (HP-85, HP-86, HP-87) a kończąc na takich systemach profesjonalnych, jak HP-9845 (System 45) i Engineering Graphics System (EGS)/45. Terminale graficzne, takie jak HP-2623, łączy się z zewnętrznym systemem komputerowym. Większość komputerów graficznych firmy Hewlett-Packard ma wbudowane monitory (HP-85 ma oddzielny monitor) i specjalne instrukcje graficzne. HP-85 ma rozdzielczość 192×256; największa rozdzielczość dla HP-87 wynosi 544×240. Profesjonalne systemy cechuje większa rozdzielczość, kolorowe monitory oraz specjalizowane oprogramowanie graficzne.

Komputery Tektronix

Serię Tektronix cechują duże możliwości graficzne. W systemach tych wykorzystuje się lampę pamięciową. System 4051 i 4052 mają rozdzielczość 1024×780, a system 4054 rozdzielczość 4096×3125. Firma Tektronix dostarcza również kilka terminali graficznych, w tym również kolorowe. Systemy graficzne mają wbudowane funkcje związane z realizacją wyrafinowanych instrukcji graficznych.



Rys. 2.5. Mikrokomputer 4052 firmy Tektronix (za zgodą Tektronix, Inc.)

Komputery Inteligent Systems

Te systemy grafiki kolorowej znane były wcześniej jako komputery Intercolor. Zawierają one mikrokomputery serii 8000 i serii 3650/9650. W systemach tych wykorzystuje się mikroprocesor 8080A. Wszystkie one mają wbudowane monitory z możliwością wyboru 8 kolorów dla planu przedniego i 8 dla tła. Standardowa rozdzielczość wynosi 160 × 192, w niektórych systemach 384 × 480. System zawiera duży zbiór instrukcji graficznych.

Komputery Chromatics

Systemy Color Graphics (CG) firmy Chromatics zawierają zarówno 8-bitowe (Z80), jak i 16-bitowe (MC68000) mikrokomputery. Te profesjonalne systemy mają wbudowane monitory, które umożliwiają równoczesne wyświetlanie do 256 kolorów. Rozdzielczość ekranu zmienia się od 512 × 512 do 1024 × 768. Systemy te mają rozbudowany zbiór instrukcji dla grafiki kolorowej.

Komputery Cromemco

Firma Cromemco Inc. produkuje system graficzny zbudowany za pomocą mikroprocesora Z80A. System ten ma wbudowany monitor i oferuje możliwość korzystania z ponad 4000 kolorów. Rozdzielczość ekranu wynosi 754×482. System zawiera rozbudowany zestaw specjalnych instrukcji graficznych.

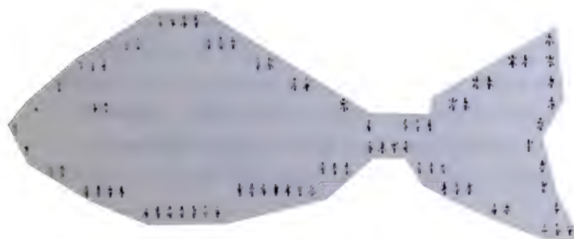
Mikrokomputerowe systemy graficzne i terminale graficzne produkuje wiele innych firm. Mały Sinclair ZX Spectrum wykorzystujący Z80A łączy się z zewnętrznym monitorem bądź z telewizorem. System ten ma 8 kolorów, rozdzielczość 256×192 i specjalne instrukcje graficzne. Firma BMC oferuje system grafiki kolorowej z mikroprocesorem Z80 i wbudowanym monitorem o rozdzielczości 640×200; system umożliwia korzystanie z 64 kolorów i odcieni. Firma NEC dysponuje ośmiokolorowym systemem graficznym (z mikroprocesorem Z80) o rozdzielczości 160×100 (wbudowany monitor). Inteligentne terminale graficzne firmy Genisco mają różne zestawy kolorów i rozdzielczości (np. 1392×1024 w G-6100). Inteligentny terminal graficzny GRAPHICUS-80 firmy Vector Automation charakteryzuje się rozbudowanym zestawem rozkazów graficznych i rozdzielczością 4096×4096.

Część II
PODSTAWY GRAFIKI
(Początki)

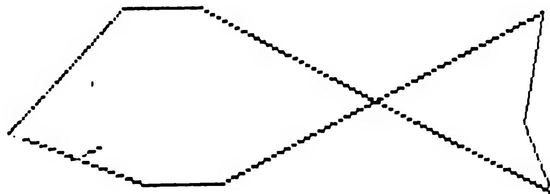
Zaczniemy od podstawowych koncepcji grafiki i metod budowy obrazu. Prosty sposób tworzenia obrazów i wykresów generowanych przez komputer polega na wykorzystaniu standardowych instrukcji wyprowadzania wyników dostępnych w języku programowania. W Basicu używa się instrukcji PRINT do tworzenia wzorów graficznych złożonych z liter, cyfr i innych symboli. Bardziej efektywne metody stosuje się w wielu systemach mikrokomputerowych, w których można korzystać ze specjalnych *instrukcji graficznych*. Instrukcje takie umożliwiają „rysowanie" obrazu złożonego z punktów i linii. Omawiając podstawowe metody skorzystamy zarówno z instrukcji PRINT, jak i z instrukcji specjalnych.

3. Proste obrazy

Obrazy można konstruować za pomocą odpowiednio rozmieszczonych znaków na ekranie. Na rysunku 3.1 przedstawiono prosty obraz utworzony przez niepołączone znaki. Takie obrazy można zazwyczaj łatwo otrzymać, korzystając z instrukcji PRINT w języku Basic. Obrazy takie są wystarczające w niektórych zastosowaniach.



Rys. 3.1. Obraz utworzony ze znaków za pomocą instrukcji PRINT



Rys. 3.2. Obraz utworzony z odcinków za pomocą specjalnych instrukcji graficznych

Ten sam obraz utworzony za pomocą ciągłych linii pokazano na rys. 3.2; użyto tu specjalnych instrukcji graficznych. W rozdziale 3 omówimy możliwości tworzenia obrazów graficznych za pomocą instrukcji PRINT, a następnie wprowadzimy instrukcje graficzne typowe dla wielu systemów mikrokomputerowych.

3.1. Rysowanie za pomocą instrukcji PRINT

Za pomocą instrukcji PRINT można wyświetlać znaki wzdłuż wiersza ekranu lub drukarki. Mogą to być dowolne litery, cyfry, znaki przestankowe i inne symbole wchodzące w skład zbioru znaków języka Basic. Na przykład po instrukcji

```
10 PRINT  "* * * *"
```

od początku wiersza będzie wyświetlony wzór złożony z czterech gwiazdek. Za pomocą instrukcji

```
10 PRINT  "****************"
20 PRINT  "*              *"
30 PRINT  "*              *"
40 PRINT  "****************"
```

możemy wyświetlić kontur pudełka. Za pomocą następujących instrukcji

```
10 PRINT  "GGGGG RRRR   AAA  PPPP  H   H I CCCCC SSSSS"
20 PRINT  "G   G R   R A   A P   P H   H I C     S    "
30 PRINT  "G     R   R A   A P   P H   H I C     SSS  "
40 PRINT  "G  GG RRRR  AAAAA PPPP  HHHHH I C      SSS "
50 PRINT  "G   G R   R A   A P     H   H I C         S"
60 PRINT  "GGGGG R   R A   A P     H   H I CCCCC SSSSS"
```

wyświetlimy powiększone litery.

Podobnie program 3.1 umożliwia wykreślenie ryby przedstawionej na rys. 3.1.

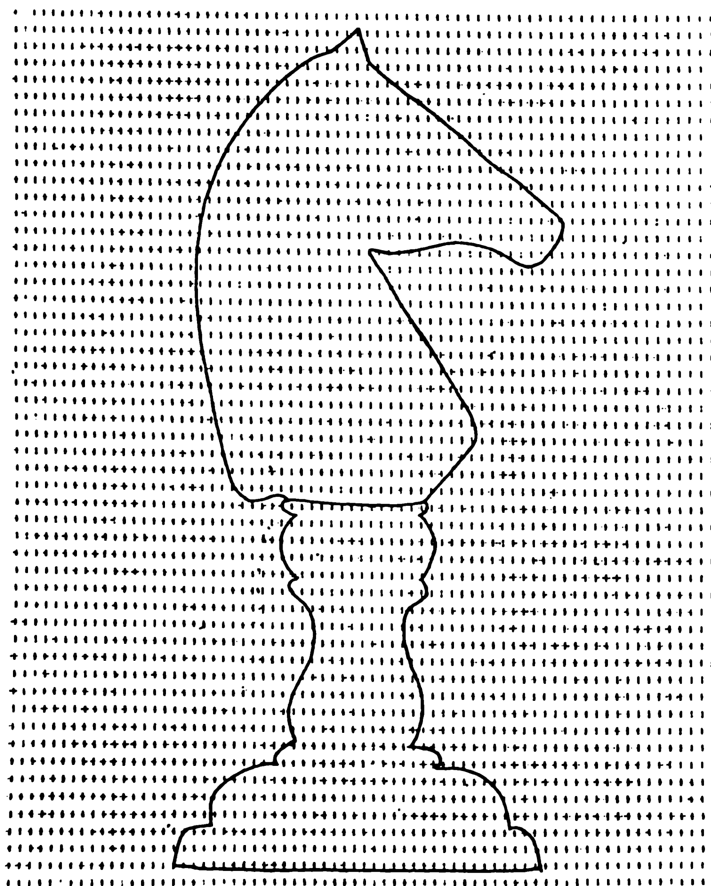
Program 3.1. Rysunek ryby wykonywany za pomocą instrukcji PRINT

```
10  'PROGRAM 3.1. Ryba utworzona za pomocą instrukcji PRINT
20  PRINT "            ****"
30  PRINT "         **     ***                        *"
40  PRINT "      ***           **                  ** *"
50  PRINT "    *                  **            **    *"
60  PRINT "  *    **                  *       **      *"
70  PRINT "*                            *  ***       *"
80  PRINT " *                           ****        *"
90  PRINT "   **                    ***     ***      *"
100 PRINT "      ****        *******         ***     *"
110 PRINT "          *******                     **   *"
120 PRINT "                                          ***"
130 END
```

Rysunki można wykonywać na papierze (przy użyciu drukarki) lub na ekranie korzystając z dostępnych wierszy. Liczba wierszy dostępnych do wyświetlania obrazów na ekranie zależy od systemu mikrokomputerowego i wynosi od 16 do 48, zaś liczba znaków w wierszu od 23 do 80. Jeżeli chcielibyśmy wyraz GRAPHICS umieścić w środku ekranu o 16 wierszach 40-znakowych, to wyraz powinien zaczynać się od pozycji 17 w wierszu 8. Jedna z metod realizacji tego zadania polega na

użyciu ciągu instrukcji PRINT do opuszczenia siedmiu wierszy, następnie zaś instrukcji PRINT TAB(17); „GRAPHICS".

Przygotowując złożony obraz do wyświetlenia można wcześniej narysować jego szkic na papierze milimetrowym. Następnie, na podstawie tego szkicu, ustala się pozycje znaków w wyświetlanym obrazie. Każdemu poziomemu wierszowi znaków na szkicu odpowiada jeden wiersz ekranu (lub wiersz drukarki). Papier milimetrowy jest pokryty siatką kwadratową, natomiast powierzchnia zajmowana przez znak na ekranie jest prostokątem, w którym często wysokość jest większa niż szerokość. Korzystając bezpośrednio ze szkicu na papierze milimetrowym przy określaniu pozycji znaków, po wyświetleniu uzyska się obraz zniekształcony. Możemy przygotować specjalny papier do szkicowania zadrukowując stronę znakami plus (+),



Rys. 3.3. Szkic rysunku na papierze z nadrukowanymi znakami plus. Szkic ten umożliwia określenie pozycji znaków dla programu 3.2

tak jak to pokazano na rys. 3.3. Obraz według szkicu przedstawionego na tym rysunku uzyskuje się za pomocą programu 3.2.

Program 3.2. Rysunek konika szachowego wykonywany za pomocą instrukcji PRINT

```
10  'PROGRAM 3.2. Konik szachowy utworzony za pomocą instrukcji PRINT
20  PRINT TAB(10);"                   #"
30  PRINT TAB(10);"                ####"
40  PRINT TAB(10);"             #########"
50  PRINT TAB(10);"           #############"
60  PRINT TAB(10);"          ################"
70  PRINT TAB(10);"        ####################"
80  PRINT TAB(10);"       #######################"
90  PRINT TAB(10);"      ##########################"
100 PRINT TAB(10);"     #############################"
110 PRINT TAB(10);"     ################################"
120 PRINT TAB(10);"    ###################################"
130 PRINT TAB(10);"    ####################################"
140 PRINT TAB(10);"   #####################################"
150 PRINT TAB(10);"   ####################################"
160 PRINT TAB(10);"   ###################             ###"
170 PRINT TAB(10);"   ####################"
180 PRINT TAB(10);"   #####################"
190 PRINT TAB(10);"   ######################"
200 PRINT TAB(10);"   #######################"
210 PRINT TAB(10);"   ########################"
220 PRINT TAB(10);"   #########################"
230 PRINT TAB(10);"    #########################"
240 PRINT TAB(10);"    #########################"
250 PRINT TAB(10);"    ##########################"
260 PRINT TAB(10);"    ###########################"
270 PRINT TAB(10);"     ###########################"
280 PRINT TAB(10);"     ##########################"
290 PRINT TAB(10);"      ########################"
300 PRINT TAB(10);"      #######################"
310 PRINT TAB(10);"        ###################"
320 PRINT TAB(10);"             #############"
330 PRINT TAB(10);"           #################"
340 PRINT TAB(10);"           #################"
350 PRINT TAB(10);"            ###############"
360 PRINT TAB(10);"             #############"
370 PRINT TAB(10);"              ###########"
380 PRINT TAB(10);"              ###########"
390 PRINT TAB(10);"              ###########"
400 PRINT TAB(10);"              ###########"
410 PRINT TAB(10);"              ###########"
420 PRINT TAB(10);"              ###########"
430 PRINT TAB(10);"            ###############"
440 PRINT TAB(10);"            ###############"
450 PRINT TAB(10);"            ###############"
460 PRINT TAB(10);"            ###############"
470 PRINT TAB(10);"        #######################"
480 PRINT TAB(10);"    ###############################"
490 PRINT TAB(10);"    ###############################"
500 PRINT TAB(10);"    ###############################"
510 PRINT TAB(10);" #####################################"
520 PRINT TAB(10);" #####################################"
530 PRINT TAB(10);" #####################################"
540 END
```

Ponieważ w programie 3.2 linie od 370 do 420 są identyczne, można zamiast nich zastosować pętlę z jedną instrukcją PRINT. Przy tworzeniu dowolnego obrazu zawierającego powtarzalne lub symetryczne wzory używa się *pętli*. Przykładowo, piramida z rys. 3.4 może być wyświetlona za pomocą programu 3.3.

```
              A
             AAA
            AAAAA
           AAAAAAA
          AAAAAAAAA
         AAAAAAAAAAA
        AAAAAAAAAAAAA
       AAAAAAAAAAAAAAA
      AAAAAAAAAAAAAAAAA
     AAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAA
   AAAAAAAAAAAAAAAAAAAAAAA
  AAAAAAAAAAAAAAAAAAAAAAAAA
 AAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Rys. 3.4. Symetryczne wzory, takie jak piramida otrzymana za pomocą programu 3.3, można tworzyć korzystając z pętli, które minimalizują liczbę instrukcji PRINT

Program 3.3. Rysunek symetrycznego wzoru (piramida) wykonywany za pomocą instrukcji PRINT i pętli programowych

```
10  'PROGRAM 3.3. Program rysowania piramidy
20  K = 30
30  FOR N = 1 TO 29 STEP 2
40    PRINT TAB(K);
50    FOR J = 1 TO N
60      PRINT "A";
70    NEXT J
80    PRINT
90    K = K - 1
100 NEXT N
110 END
```

Liczbę instrukcji PRINT potrzebnych do uzyskania obrazu można również zmniejszyć wtedy, kiedy pozycje, które należy drukować, są *zakodowane* w instrukcjach DATA. Te zakodowane dane mogą być potem odczytane z instrukcji DATA, zdekodowane i wydrukowane. Poszczególne wiersze można kodować wieloma sposobami. Na przykład każdy wiersz obrazu może być zapamiętany w postaci *pary liczb*. Pierwsza liczba określa *początkową pozycję* drukowania, druga zaś liczbę znaków,

Program 3.4. Rysunek konika szachowego wykonywany za pomocą instrukcji PRINT i zakodowanych danych

```
10  'PROGRAM 3.4. Konik szachowy. Metoda zakodowanych danych
20       'Każda drukowana linia jest pamiętana jako para liczb.
30       'Pierwsza liczba określa pozycję początku drukowania.
40       'Druga liczba określa,ile znaków należy drukować.
50       'Wprowadzanie kończy się po wczytaniu 0,0.
60       'W wierszu 110 mozna zmienić drukowany znak.
70  READ P, N
80  IF P = 0 AND N = 0 THEN 200
90  PRINT TAB(P);
100 FOR K = 1 TO N
110   PRINT "#";
120 NEXT K
130 GOTO 70
140 DATA 28,1,25,5,22,9,20,13,19,16,17,20,16,23,15,26,14,29,14
150 DATA 32,13,35,13,36,12,37,12,36,12,19,44,3,12,20,12,21,12,22
160 DATA 12,23,12,24,12,25,13,25,13,26,13,27,14,27,14,26,15,24,15
170 DATA 23,17,19,22,13,20,17,20,17,21,15,22,13,23,11,23,11,23,11
180 DATA 23,11,23,11,23,11,21,15,21,15,21,15,21,15,17,23,13,31,13
190 DATA 31,13,31,10,37,10,37,10,37,0,0
200 END
```

którą należy wydrukować w wierszu. Taki sposób kodowania wykorzystano w programie 3.4 generującym obraz konika szachowego z rys. 3.3.

Korzystając z instrukcji PRINT można uzyskać efekt *cieniowania* obrazu. Realizuje się to używając różnych znaków do tworzenia obrazu. Znaki takie jak *, & i inne będą tworzyły ciemniejsze powierzchnie niż znaki 0 i 1. Niektóre drukarki umożliwiają uzyskanie efektu stopniowania cieniowania przez *nadrukowywanie*, tj. umieszczanie dwóch lub więcej znaków na tej samej pozycji.

Inny sposób zmieniania tekstury obrazów polega na stosowaniu specjalnych znaków graficznych dostępnych w niektórych mikrokomputerach. W zależności od systemu znaki te mają różne kształty i wzory. Mogą występować na klawiaturze i mogą być wykorzystywane przez instrukcje PRINT, tak jak inne znaki. W innych systemach można wyświetlać znaki specjalne za pomocą rozkazu PRINT CHR$(A), przy czym A jest kodem ASCII żądanego znaku. Na rysunku 3.5 pokazano obraz utworzony z takich znaków graficznych.

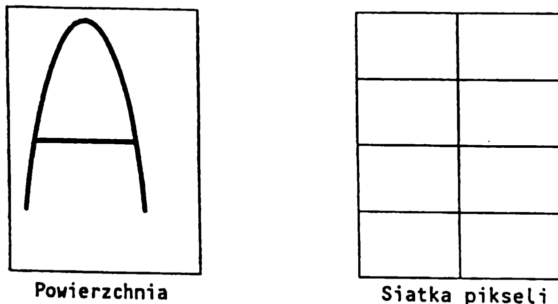


Rys. 3.5. Obraz utworzony za pomocą specjalnych znaków graficznych

3.2. Zasady grafiki punktowej

Każdy znak wyświetlony na ekranie monitora zajmuje mały prostokąt. W wielu mikrokomputerach powierzchnia ta jest dzielona na jeszcze mniejsze prostokąty. Te mniejsze prostokąty są nazywane *elementami obrazu* lub *pikselami*, lub po prostu *punktami*. Zależnie od systemu liczba pikseli zajmowanych przez jeden znak wynosi

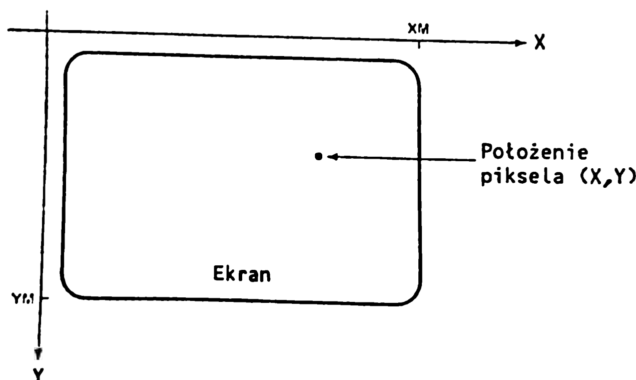
od 2 do 8 punktów w poziomie i od 3 do 12 w pionie. Na rysunku przedstawiono *siatkę* 2×4 pikseli odpowiadającą powierzchni zajmowanej przez znak. Monitor stosowany w takim systemie będzie miał w poziomie dwa razy więcej punktów niż znaków, w pionie zaś cztery razy więcej punktów niż znaków. Gdyby taki system miał 48 wierszy po 80 znaków w wierszu, moglibyśmy narysować punkt w dowolnej ze 160 pozycji w wierszu ekranu i dowolnej ze 192 pozycji w pionie.



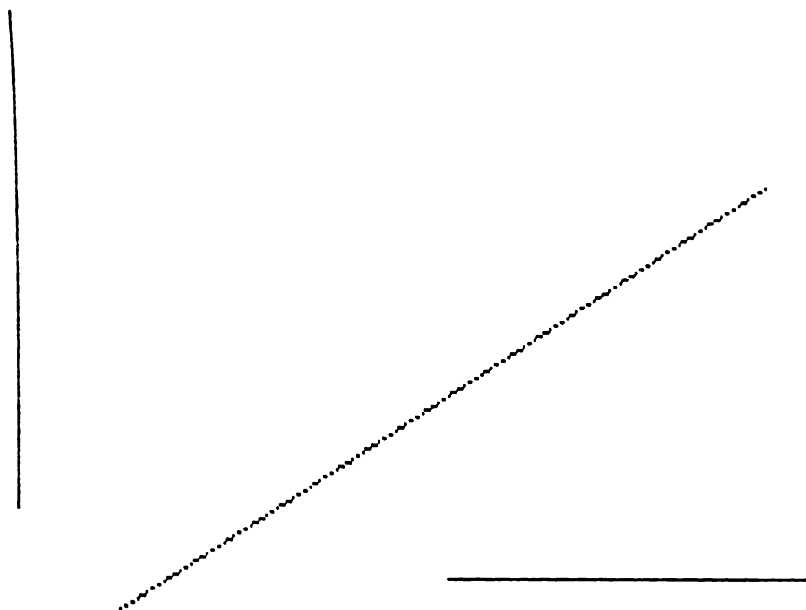
Rys. 3.6. Powierzchnia zajmowana przez znak podzielona na siatkę mniejszych prostokątnych pikseli

W *trybie graficznym* piksele są dostępne bezpośrednio. W niektórych systemach przejście do trybu graficznego wymaga napisania specjalnej instrukcji inicjującej, takiej jak GR lub SCREEN. Inne systemy automatycznie przechodzą do trybu graficznego po rozpoznaniu dowolnej ze specjalnych instrukcji graficznych, takich jak „narysuj punkt" lub „narysuj linię".

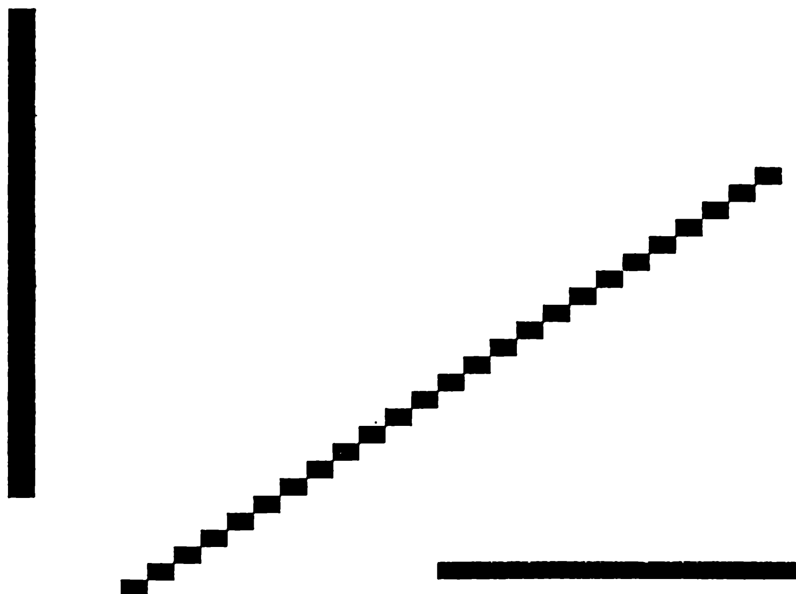
Rysowanie punktu na ekranie monitora oznacza, że żądamy od komputera, by „włączył" on małe prostokątne światło we wskazanej pozycji piksela. Pozycję jednego piksela określają jego współrzędne w postaci pary liczb całkowitych (X, Y).



Rys. 3.7. W wielu mikrokomputerach początek układu współrzędnych znajduje się w lewym górnym rogu ekranu



(a)



(b)

Rys. 3.8. Odcinki narysowane z dużą rozdzielczością (a) i z mniejszą rozdzielczością (b)

Liczba X określa odległość w poziomie, Y zaś odległość w pionie od początku układu współrzędnych. W wielu mikrokomputerach odległości te są mierzone od lewej do prawej strony ekranu i z góry na dół. Oznacza to, że początek układu współrzędnych znajduje się w lewym górnym rogu (rys. 3.7). Współrzędna pozioma X może przyjmować wartości (licząc od lewej strony) 0, 1, 2, ... do pewnej wartości maksymalnej XM. Współrzędna pionowa Y zmienia się przy ruchu z góry na dół, przyjmując wartości 0, 1, 2, ..., YM. Dla takiego układu współrzędnych piksel w górnym lewym rogu ekranu ma współrzędne (0, 0), a piksel w dolnym prawym rogu (XM, YM). Taki układ współrzędnych jest stosowany w mikrokomputerach Apple, TRS-80, IBM i ATARI. Wszystkie nasze dyskusje o grafice będą odnosiły się do takiego układu współrzędnych.

W niektórych mikrokomputerach przyjęto bardziej konwencjonalny układ współrzędnych z początkiem w dolnym lewym rogu ekranu. W tych systemach piksel w dolnym lewym rogu ekranu ma współrzędne (0, 0), a piksel w górnym prawym rogu (XM, YM). Przykładami systemów, w których początek układu współrzędnych jest w dolnym lewym rogu ekranu, są systemy Hewlett-Packard, Tektronix i Intecolor. Nasze przykładowe programy mogą być przeniesione do tego układu współrzędnych przez zastąpienie każdej wartości współrzędnej Y rysowanego punktu wartością YM – Y.

Maksymalne wartości szerokości XM i wysokości YM ekranu są od ok. 40 do kilkuset, chociaż spotyka się znacznie większe wartości. Przy większych wartościach XM i YM możemy rysować bardziej gładkie linie, co wynika z tego, że każdy punkt zajmuje mniejszą powierzchnię na ekranie. Liczba pikseli wzdłuż linii jest określana jako rozdzielczość systemu. Wpływ większej i mniejszej rozdzielczości na wygląd linii prostej zilustrowano na rys. 3.8. Linie poziome i pionowe będą gładkie niezależnie od rozdzielczości. Dla linii ukośnych im mniejsza jest rozdzielczość, tym wyraźniejszy jest „schodkowy" kształt linii.

Bardziej precyzyjne określenie rozdzielczości polega na podaniu liczby pikseli, które można narysować na długości 1 cm. Dlatego większy ekran, na którym można narysować w poziomie tyle samo pikseli co na mniejszym ekranie, będzie miał mniejszą rozdzielczość (mniej punktów na centymetr). Z większością mikrokomputerów mogą współpracować monitory o różnych wymiarach. Jednak zmiana wymiarów ekranu nie zmienia liczby pikseli generowanych przez komputer. Liczba ta jest określona przez parametry graficzne ustalone w czasie projektowania systemu. W niektórych systemach jest możliwy wybór rozdzielczości. Takie tryby pracy są często określane jako: mała rozdzielczość (mało punktów), średnia rozdzielczość (więcej punktów) i duża rozdzielczość (najwięcej punktów).

3.3. Rysowanie punktów

Chcąc narysować punkt musimy dysponować odpowiednią instrukcją graficzną. W przeciwieństwie do instrukcji PRINT instrukcje graficzne nie są standaryzowane. W różnych systemach mikrokomputerowych używa się różnych instrukcji do wykonania tych samych operacji. Ponieważ podstawowe operacje w grafice (wybranie rozdzielczości, rysowanie punktów itd.) są takie same, możemy omawiać te operacje korzystając ze zbioru hipotetycznych instrukcji graficznych. Ten zbiór będzie przez nas używany we wszystkich dalszych dyskusjach. Związki między hipotetycznymi rozkazami a rzeczywistymi instrukcjami w określonym systemie przedstawiono w dodatku A.

W niektórych systemach musimy przechodzić do trybu graficznego i określać pożądaną rozdzielczość. W naszym zbiorze hipotetycznych instrukcji dopuszczamy tylko jedną rozdzielczość, natomiast instrukcja inicjująca tryb graficzny jest zdefiniowana następująco:

GRAPHICS — przejście do trybu graficznego

Będąc już w trybie graficznym możemy narysować punkt za pomocą instrukcji

POINTPLOT X, Y — Umieszcza piksel na ekranie w punkcie o współrzędnych X, Y; X i Y mogą być stałymi numerycznymi lub wyrażeniami. Wartości niecałkowite są zaokrąglane.

W niektórych systemach instrukcje rysowania punktu powodują obcinanie do wartości całkowitych zamiast zaokrąglania. Przykład użycia instrukcji GRAPHICS i POINTPLOT podano w programie 3.5. Program ten rysuje punkt w miejscu określonym w instrukcji.

Program 3.5. Rysowanie punktu

```
10  'PROGRAM 3.5. Rysowanie pojedynczego punktu
20  PRINT "Wprowadź współrzędne piksela jako parę liczb całkowitych"
30  INPUT X, Y
40  GRAPHICS
50  POINTPLOT X,Y
60  END
```

Nie powinniśmy podejmować prób umieszczania punktu poza granicami ekranu. Mogłoby to w rezultacie spowodować albo błąd w wykonaniu programu, albo zmienić położenie punktu wskutek *efektu zawijania*. Efekt ten pojawia się wówczas, gdy punkt rysowany poza granicami ekranu z jednej strony „zawija się" i pojawia z drugiej strony ekranu. Aby uniknąć takich problemów, współrzędne poziome i pionowe nigdy nie powinny być ujemne ani nie powinny przekraczać wartości maksymalnych dla ekranu. W celu zabezpieczenia się przed próbą rysowania poza ekranem możemy do naszego programu wprowadzić testy

$$0 \leqslant X \leqslant XM$$
$$0 \leqslant Y \leqslant YM \qquad (3.1)$$

W tych testach XM jest maksymalną dopuszczalną pozycją piksela w poziomie, YM zaś maksymalną dopuszczalną pozycją piksela w pionie. Program 3.6 prezentuje sposób użycia tych testów w przypadku ekranu o wymiarach 280×160 pikseli.

Program 3.6. Testowanie, czy punkt mieści się na ekranie, rysowanie punktu

```
10  'PROGRAM 3.6. Test na granicach ekranu i rysowanie punktu
20  PRINT "Podaj współrzędne piksela jako parę liczb całkowitych"
30  INPUT X, Y
40  IF X < 0 OR X > 279 OR Y < 0 OR Y > 159 THEN 80
50  GRAPHICS
60  POINTPLOT X,Y
70  GOTO 90
80  PRINT "Próba rysowania poza ekranem"
90  END
```

W wielu zastosowaniach dogodnie jest mieć możliwość usunięcia punktu wcześniej narysowanego. Aby to zrealizować, wprowadzamy następującą instrukcję:

> POINTOFF X, Y — Usuwa wcześniej narysowany piksel w pozycji X, Y; X i Y mogą być stałymi numerycznymi lub wyrażeniami. Wartości niecałkowite są zaokrąglane.

Jeżeli w pozycji X, Y nie jest wyświetlany piksel, to instrukcja POINTOFF nie spowoduje żadnego widocznego efektu. Program 3.7 pokazuje, w jaki sposób można użyć tej instrukcji, aby usunąć piksel po opóźnieniu ok. 2 s.

Program 3.7. Rysowanie i usuwanie punktu

```
10  'PROGRAM 3.7. Wyświetlanie i usuwanie punktu
20  PRINT "Podaj wartości maksymalne poziomą i pionową"
30  PRINT "dla przyjętej rozdzielczości"
40  INPUT XM, YM
50  PRINT "Podaj współrzędne X i Y rysowanego punktu"
60  INPUT X, Y
70  IF X < 0 OR X > XM OR Y < 0 OR Y > YM THEN 130
80  GRAPHICS
90  POINTPLOT X,Y
100 FOR J = 1 TO 1000: NEXT J
110 POINTOFF X,Y
120 GOTO 140
130 PRINT "Próba rysowania poza ekranem"
140 END
```

Pętle opóźnienia (patrz instrukcja 100 w programie 3.7) można umieszczać w programach graficznych w celu uzyskania efektu migotania punktu. Pętle można również stosować, aby utrzymać dany obraz, zanim zostanie utworzony następny. Opóźnienie uzyskane za pomocą takich pętli wynosi ok. 1 s na każde 500 iteracji pętli, zależnie od szybkości przetwarzania w konkretnym systemie.

Często chcemy wymazać (wyzerować) cały ekran, zanim będziemy tworzyli obraz lub wyświetlali program. W tym celu definiujemy instrukcję:

CLEARSCREEN — Wymazuje ekran i cofa kursor do górnego lewego rogu ekranu.

Instrukcja ta może być używana w trybie graficznym lub w trybie tekstowym. Stosowanie instrukcji CLEARSCREEN ilustruje program 3.8 rysujący przypadkowy zbiór punktów. Funkcja RND, używana w tym programie, przyjmuje różne postaci w różnych wersjach Basicu.

Program 3.8. Rysowanie wzoru złożonego z punktów losowych

```
10  'PROGRAM 3.8. Rysowanie przypadkowych punktów
20  PRINT "Podaj wartości maksymalne poziomą i pionową"
30  PRINT "dla przyjętej rozdzielczości"
40  INPUT XM,YM
50  PRINT "Podaj liczbę rysowanych punktów"
60  INNPUT P
70  CLEARSCREEN
80  GRAPHICS
90  FOR K = 1 TO P
100   X = XM * RND(0)
110   Y = YM * RND(0)
120   POINTPLOT X,Y
130 NEXT K
140 END
```

3.4. Rysowanie odcinków

Instrukcja POINTPLOT może być używana do rysowania odcinków. W celu uzyskania odcinka równoległego do jednej z osi rysujemy wszystkie możliwe punkty wzdłuż prostej między dwoma końcami odcinka. Odległość między kolejnymi pikselami jest jednostkowa. Instrukcja POINTPLOT narysuje np. odcinek pionowy, gdy współrzędna Y będzie kolejno zwiększana o 1, a wartość współrzędnej X pozo-

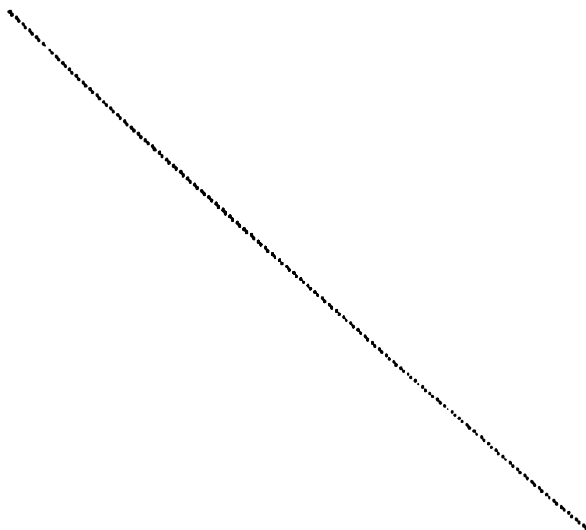
Program 3.9. Rysowanie pionowej linii metodą punktową

```
10  'PROGRAM 3.9. Rysowanie linii pionowej
20  CLEARSCREEN
30  PRINT "Podaj wartości maksymalne poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM,YM
60  PRINT "Podaj współrzędną X linii pionowej"
70  INPUT X
80  IF X >= 0 AND X <= XM THEN 110
90  PRINT "Współrzędna X poza granicami ekranu. Spróbuj jeszcze raz."
100 GOTO 60
110 PRINT "Podaj współrzędne Y punktów końcowych"
120 INPUT Y1,Y2
130 IF Y1 >= 0 AND Y1 <= YM AND Y2 >= 0 AND Y2 <= YM THEN 160
140 PRINT "Współrzędna Y poza granicami ekranu. Spróbuj jeszcze raz."
150 GOTO 110
160 CLEARSCREEN
170 GRAPHICS
180 FOR Y = Y1 TO Y2
190   POINTPLOT X,Y
200 NEXT Y
210 END
```

stanie stała. Przykładami rysowania odcinków poziomych i pionowych są programy 3.9 i 3.10. Programy te kreślą odcinki ciągłe, jeśli kolejne piksele są rysowane w postaci małych prostokątów stycznych z pikselami narysowami poprzednio.

Program 3.10. Rysowanie poziomej linii metodą punktową

```
10  'PROGRAM 3.10. Rysowanie linii poziomej
20  CLEARSCREEN
30  PRINT "Podaj wartości maksymalne poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM,YM
60  PRINT "Podaj współrzędną Y linii poziomej"
70  INPUT Y
80  IF Y >= 0 AND Y <= YM THEN 110
90  PRINT "Współrzędna Y poza granicami ekranu. Spróbuj jeszcze raz."
100 GOTO 60
110 PRINT "Podaj współrzędne X punktów końcowych"
120 INPUT X1,X2
130 IF X1 >= 0 AND X1 <= XM AND X2 >= 0 AND X2 <= XM THEN 160
140 PRINT "Współrzędna X poza granicami ekranu. Spróbuj jeszcze raz."
150 GOTO 110
160 CLEARSCREEN
170 GRAPHICS
180 FOR X = X1 TO X2
190   POINTPLOT X,Y
200 NEXT X
210 END
```



Rys. 3.9. Linia prosta o nachyleniu M = 1 i przesunięciu B = 0 narysowana za pomocą programu 3.11

Rysowanie odcinków ukośnych jest trudniejsze. Żeby wykreślić odcinek ukośny za pomocą instrukcji POINTPLOT, musimy obliczać wartości współrzędnych wzdłuż odcinka. W tym celu korzystamy z zależności (3.2) wiążącej wartości X i Y dla linii prostej

$$Y = M * X + B \tag{3.2}$$

W tym równaniu M jest *współczynnikiem nachylenia* prostej przyjmującym wartości dodatnie, ujemne bądź zero. Przy M = 0 otrzymamy prostą poziomą, a przy bardzo dużych wartościach M proste prawie pionowe. Współczynnik B jest wartością Y przy X = 0 i jest zwany *przesunięciem* linii. Przy B = 0 prosta przechodzi przez początek układu współrzędnych.

Program 3.11. Rysowanie prostej metodą punktową z wykorzystaniem równania prostej

```
10  'PROGRAM 3.11. Rysowanie prostej z pomocą instrukcji POINTPLOT
20  CLEARSCREEN
30  PRINT "Podaj wartości maksymalne poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM,YM
60  PRINT "Podaj wartości nachylenia i przesunięcia"
70  INPUT M, B
80              'Jeżeli wartości nachylenia i przesunięcia są ujemne,
90              'to linia jest poza ekranem.
100 IF M < 0 AND B < 0 THEN PRINT "Linia poza ekranem":GOTO 430
110             'Jeżeli nachylenie jest dodatnie i przesunięcie
120             'większe od YM,to linia jest poza ekranem.
130 IF M > 0 AND B > YM THEN PRINT "Linia poza ekranem":GOTO 430
140             'W przeciwnym przypadku znajdź skrajny lewy punkt linii.
150             'Jeżeli B jest zawarte między 0 a YM,to skrajny lewy
160             'punkt leży na lewym brzegu ekranu.
170 X1 = 0
180             'Jeżeli nachylenie jest ujemne i przesunięcie jest
190             'większe od YM,to skrajny lewy punkt leży na dolnym
200             'brzegu ekranu (gdzie Y = YM) i X1 = (YM - B) / M.
210 IF M < 0 AND B > YM THEN X1 = (YM - B) / M
220             'Jeżeli skrajny lewy punkt jest poza XM,to linia jest
225             'poza ekranem.
230 IF X1 > XM THEN PRINT "Linia poza ekranem":GOTO 430
240             'Jeżeli nachylenie jest dodatnie i przesunięcie jest
250             'ujemne,to skrajny lewy punkt leży na górnej krawędzi
260             'ekranu (gdzie Y = 0) i X1 = (0 - B) / M lub X1 = -B / M.
270 IF M > 0 AND B < 0 THEN X1 = -B / M
280             'Jeżeli ten punkt jest poza XM,to linia jest poza ekranem
290 IF X1 > XM THEN PRINT "Linia poza ekranem":GOTO 430
300             'W przeciwnym przypadku linia przynajmniej częściowo jest
310             'na ekranie. Zacznij w skrajnym lewym punkcie linii (X1).
320             'Zwiększając wartość X obliczaj nowe wartości Y i rysuj
325             'X,Y. Kontynuuj aż:
330             '1.  X > XM (linia poza prawym brzegiem ekranu)
340             '2.  Y < 0  (linia poza górnym brzegiem ekranu) lub
350             '3.  Y > YM (linia poza dolnym brzegiem ekranu)
360 CLEARSCREEN
370 GRAPHICS
380 FOR X = X1 TO XM
390   Y = M * X + B
400   IF Y < 0 OR Y > YM THEN 430
410   POINTPLOT X,Y
420 NEXT X
430 END
```

Korzystając z instrukcji POINTPLOT i równania (3.2) można napisać program realizujący ogólny algorytm rysowania odcinka. Program 3.11 umożliwia rysowanie odcinka dla zadanych wartości M i B. Program ten najpierw określa, czy odcinek może być narysowany na ekranie. Jeżeli żaden fragment odcinka nie może być narysowany w obrębie ekranu, program po prostu drukuje taki komunikat. Jeżeli część odcinka może być narysowana, program 3.11 rysuje tę widzialną część odcinka od jednego brzegu ekranu do drugiego. Na rysunku 3.9 pokazano wynik działania programu 3.11 dla przypadku M = 1. Przedstawiony odcinek wygląda inaczej, niż spodziewalibyśmy się rysując odcinek w konwencjonalnym układzie współrzędnych, kiedy to dla M dodatnich linia wznosi się od lewej do prawej strony. Efekt ten jest skutkiem przyjęcia położenia punktu (0, 0) w lewym górnym rogu ekranu. Chcąc rysować odcinki o przeciwnym nachyleniu, możemy zmodyfikować program mnożąc każdą wprowadzaną wartość nachylenia przez −1 i zmieniając każdą wprowadzaną wartość przesunięcia B na YM − B.

Jeżeli wartość nachylenia jest większa niż 1, rysowane odcinki nie będą ciągłe. W takim przypadku obserwujemy przerwy między rysowanymi pikselami. Można to korygować zwiększając o jeden współrzędną Y nie zaś współrzędną X, gdy ABS(M) > 1. Program można również tak zmodyfikować, aby odcinek był rysowany w środku ekranu. Metody realizacji tego przedyskutujemy w rozdz. 4 przy omawianiu wykresów.

Jeżeli chcemy narysować odcinek między punktami końcowymi o podanych współrzędnych, musimy określić współczynnik nachylenia M i wartość przesunięcia B. Jest to potrzebne do obliczania punktów wzdłuż linii. Oznaczając współrzędne końców linii jako (X1, Y1) i (X2, Y2), możemy obliczyć wartości M i B z zależności:

$$M = (Y2 - Y1)/(X2 - X1)$$
$$B = Y1 - M * X1 \qquad (3.3)$$

Program rysowania odcinków wykorzystujący równania (3.3), po określeniu współrzędnych końców odcinka, obliczałby wartości M i B, a następnie rysowałby punkty wzdłuż odcinka między punktami końcowymi.

W wielu systemach mikrokomputerowych stosuje się instrukcję graficzną, która automatycznie rysuje odcinek między dwoma zadanymi punktami. Do realizacji takiego zadania definiujemy następującą instrukcję:

> DRAWLINE X1, Y1 TO X2, X2 — Rysuje odcinek od punktu (X1, Y1) do punktu (X2, Y2). X1, Y1, X2, Y2 mogą być stałymi numerycznymi lub wyrażeniami. Wartości niecałkowite są zaokrąglane.

W programie 3.12 instrukcja DRAWLINE jest wykorzystana do rysowania odcinka między dwoma określonymi punktami.

Program 3.12. Rysowanie odcinka z wykorzystaniem instrukcji DRAWLINE

```
10  'PROGRAM 3.12. Rysowanie prostej za pomocą instrukcji DRAWLINE
20  CLEARSCREEN
30  PRINT "Podaj wartości maksymalne poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM,YM
60  PRINT "Podaj współrzędne pierwszego punktu"
70  INPUT X1,Y1
80  IF X1 >= 0 AND X1 <= XM THEN 110
90  PRINT "Współrzędna X poza granicami. Popraw"
100 GOTO 60
110 IF Y1 >= 0 AND Y1 <= YM THEN 140
120 PRINT "Współrzędna Y poza granicami. Popraw"
130 GOTO 60
140 PRINT "Podaj współrzędne następnego punktu"
150 INPUT X2,Y2
160 IF X2 >= 0 AND X2 <= XM THEN 190
170 PRINT "Współrzędna X poza zakresem. Popraw"
180 GOTO 140
190 IF Y2 >= 0 AND Y2 <= YM THEN 220
200 PRINT "Współrzędna Y poza zakresem. Popraw"
210 GOTO 140
220 CLEARSCREEN
230 GRAPHICS
240 DRAWLINE X1,Y1 TO X2,Y2
250 END
```

3.5. Rysowanie obrazów i kolor

Instrukcje rysowania punktów i odcinków są podstawowymi narzędziami tworzenia obrazów graficznych. Za pomocą tych instrukcji możemy konstruować kontury figur, szkielety brył, trójwymiarowe i złożone sceny. Dodatkowo, w wielu systemach graficznych istnieje możliwość poprawienia jakości obrazów dzięki użyciu koloru.

Kolor

Możliwość dodania koloru do obrazu może być cenną cechą systemu graflzycznego. Dzięki kolorowi obrazy mogą być bardziej zrozumiałe. W złożonych obrazach wyróżnia się kolorem różne obiekty i powierzchnie. Kolor wykorzystuje się do uwypuklenia ważnych części obrazu bądź do podkreślenia jego realizmu. Kolor bardzo uatrakcyjnia tworzone obrazy. W niektórych zastosowaniach, jak na przykład przy projektowaniu wzorów tkanin, nie sposób obejść się bez kolorów.

Do naszego zbioru hipotetycznych instrukcji dodajemy następującą instrukcję graficzną:

COLOR F, B — Ustala kolor piksela wyświetlanego przez następne instrukcje jako kolor F (przedni plan) na ekranie o kelorze B (tło). F i B mogą być stałymi numerycznymi lub wyrażeniami. Wartość niecałkowite będą zaokrąglone.

Kody kolorów dla F i B:

0 — czarny	4 — zielony
1 — biały	5 — żółty
2 — niebieski	6 — jasnoniebieski
3 — czerwony	7 — pomarańczowy

W programie można używać kilku instrukcji COLOR. Na przykład w wyniku następującej sekwencji instrukcji uzyska się jasnoniebieskie tło ekranu, na którym będą wyświetlone wzory w kolorach czerwonym, zielonym i żółtym:

```
COLOR 3,6
  :
COLOR 4,6
  :
COLOR 5,6
```

Obrazy wyświetlone po instrukcji zadającej takie same kolory tła i przedniego planu (F = B) są niewidoczne. Podanie równych wartości F i B, a następnie ponowne narysowanie linii widocznej na ekranie „wymazuje" ją. Jeżeli w programie nie ma instrukcji koloru, będziemy przyjmowali, że system ma standardowy zestaw kolorów, na przykład biały na czarnym tle.

W dodatku A podano odpowiedniki instrukcji COLOR F, B w różnych systemach. Liczba kolorów w różnych systemach może zmieniać się od 2 (czarny i biały) do ok. 20. W niektórych systemach liczba palet dostępnych kolorów jest duża, w innych systemach liczba kombinacji kolorów jest ustalona. Niekiedy palety kolorowe mogą być używane poza trybem graficznym do nadawania kolorów znakom.

Wyboru kombinacji kolorów dla określonego obrazu należy dokonać bardzo starannie. Przy przypadkowym wyborze można otrzymać rażące, nieprzyjemne efekty. Zazwyczaj najlepiej jest używać niewielu kolorów w obrazie. Dobre efekty uzyskuje się wybierając jako tło kolor będący „uzupełnieniem" jednego z kolorów przedniego planu. Kolorami uzupełniającymi są: czerowny i niebieskozielony, niebieski i pomarańczowy, żółty i purpurowoniebieski, zielony i magenta (czerwonopurpurowy). Dobre efekty otrzymuje się łącząc jasne tło (powiedzmy niebieskie) z ciemniejszymi kolorami. Jeżeli w obrazie będzie wiele kolorów, najlepiej wybrać tło szare lub neutralne. Efekt braku harmonii między kolorami przy dużej ich liczbie można zmniejszyć, stosując czarne obwódki wokół obszarów o różnych kolorach.

Rysowanie obrazów

W programach 3.13 do 3.18 podano przykłady stosowania instrukcji POINTPLOT, DRAWLINE i COLOR do tworzenia obrazów. Program 3.13 umożliwia rysowanie trójkąta, prostokąta lub wielokąta przy różnych kombinacjach kolorów. Współrzędne wierzchołków figury wprowadza się w kolejności, w jakiej

mają być połączone i zapamiętane w tablicach X i Y. Wielokąt jest wyświetlany przez rysowanie linii od punktu X(1), Y(1) do punktu X(2), Y(2), a następnie do punktu X(3), Y(3) itd.

Program 3.13. Rysowanie wieloboku i kolorowanie go

```
10  'PROGRAM 3.13. Rysowanie wielokąta w kolorach
20       'Program rysowania wielokąta o co najwyżej 10 bokach. Współ-
30       'rzędne wierzchołków są wprowadzane w kolejności łączenia.
40       'Wielokąt jest wyświetlany dla zadanej kombinacji kolorów
50       'wnętrza wielokąta i tła z opóźnieniem czasowym między
60       'kolejnymi obrazami.
70       '**************************************************************
80  CLEARSCREEN
90  DIM X(10), Y(10)
100 PRINT "Podaj współrzędne maksymalne poziomą i pionową"
110 INPUT XM,YM
120 PRINT "Ile wierzchołków ma wielokąt?"
130 INPUT N
140 PRINT "Podaj współrzędne wierzchołków w kolejności ich łączenia"
150 FOR K = 1 TO N
160   INPUT X(K), Y(K)
170   IF X(K) >= 0 AND X(K) <= M THEN 200
180   PRINT "Współrzędna X poza zakresem. Spróbuj jeszcze raz"
190   GOTO 160
200   IF Y(K) >= 0 AND Y(K) <= YM THEN 230
210   PRINT "Współrzędna Y poza zakresem. Spróbuj jeszcze raz"
220   GOTO 160
230 NEXT K
240 GRAPHICS
250 FOR F = 0 TO 7
260   FOR B = 0 TO 7
270     COLOR F,B
280     CLEARSCREEN
290     FOR K = 1 TO N - 1
300       DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
310     NEXT K
320     'Połącz ostatni wierzchołek z pierwszym
330     DRAWLINE X(N),Y(N) TO X(1),Y(1)
340     PRINT "W celu zmiany kolorów naciśnij dowolny klawisz"
350     IF INKEY$="" THEN 350
360   NEXT B
370 NEXT F
380 END
```

W programie 3.13 pokazano również inny rodzaj opóźnienia czasowego. W tym przypadku wykonywanie programu zatrzymuje się na czas nieokreślony w linii 350. Występująca w niektórych wersjach Basicu funkcja INKEY$ jest inicjowana przez system jako pusty łańcuch. W czasie realizacji przyjmuje ona wartość określoną przez naciśnięty przycisk klawiatury, jeżeli jakiś przycisk został naciśnięty. Taki rodzaj opóźnienia jest szczególnie użyteczny, gdy zadaniem programu jest tworzenie szeregu obrazów, wykorzystywanych np. w czasie prelekcji. Często pomocne jest włączenie mechanizmu „zachęty" razem z opóźnieniem czasowym, zwłaszcza gdy piszemy program, który będzie używany przez innych. W programie 3.13 w celu zaznaczenia, że program jest w stanie oczekiwania, przed wierszem 350 włączyliśmy komunikat „W celu zmiany kolorów naciśnij dowolny klawisz". Aby

nie drukować komunikatu na obrazie, możemy po prostu wyświetlać te instrukcje raz na początku wykonania programu. Wtedy program może wyzerować ekran, narysować obraz i czekać na reakcję użytkownika.

Program 3.14. Malowanie kolorowych prostokątów

```
10  'PROGRAM 3.14. Malowanie kolorowych prostokątów
20  CLEARSCREEN
30  PRINT "Podaj wartości maksymalne poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM,YM
60  PRINT "Podaj współrzędne górnego lewego rogu"
70  INPUT X1,Y1
80  IF X1 >= 0 AND X1 <= XM AND Y1 >= 0 AND Y1 <= YM THEN 110
90  PRINT "Współrzędne poza zakresem.Powtórz"
100 GOTO 60
110 PRINT "Wprowadź współrzędne dolnego prawego rogu"
120 INPUT X2,Y2
130 IF X2 >= 0 AND X2 <= XM AND Y2 >= 0 AND Y2 <=YM THEN 160
140 PRINT "Współrzędne poza zakresem.Powtórz"
150 GOTO 110
160 PRINT "Wprowadź kody kolorów tła i przedniego planu"
170 INPUT F,B
180 IF F >= 0 AND F <= 7 AND B >= 0 AND B <= 7 THEN 210
190 PRINT "Niedostępny kolor"
200 GOTO 160
210 CLEARSCREEN
220 GRAPHICS
230 COLOR F,B
240 FOR Y = Y1 TO Y2
250   DRAWLINE X1,Y TO X2,Y
260 NEXT Y
270 END
```

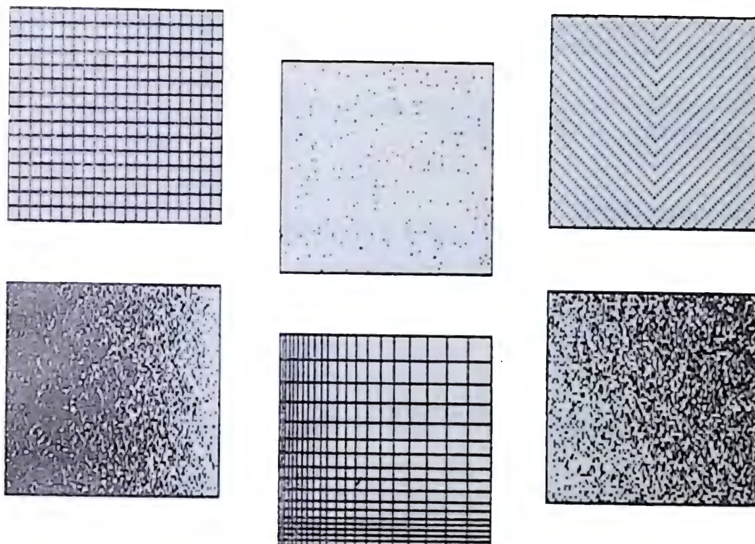
Metodę generowania kolorowych powierzchni ilustruje program 3.14. W niektórych systemach realizację tej funkcji zapewniają dodatkowe opcje graficzne; również programy rysowania linii umożliwiają uzyskanie specjalnych efektów, takich jak „malowanie". W tym programie wybieramy dowolną powierzchnię prostokątną na ekranie, określając przeciwległe wierzchołki prostokąta.

Kolorowe powierzchnie otrzymuje się rysując linie poziome, pionowe lub ukośne. W celu uzyskania specjalnych efektów można zmieniać kolejność rysowania linii i kolory. Na przykład można wypełniać powierzchnię prostokąta liniami poziomymi o losowo generowanych kolorach bądź też rysować na przemian od góry i od dołu linie kończące się w środku.

Cieniowanie obrazów uzyskuje się przez częściowe wypełnianie powierzchni, na przykład przez rysowanie co drugiej lub co trzeciej linii. Można by również rysować linie w dwóch przecinających się kierunkach, co przy zachowaniu odpowiednich odstępów dałoby efekt cieniowania krzyżowego. Inna możliwość polega na zmienianiu odstępów między liniami cieniującymi; można w ten sposób otrzymać stopniowe przejście od odcienia ciemnego do jasnego (lub od jasnego do ciemnego). Odstępy między liniami mogą być zwiększane powoli o 1 lub 2; można by również zmieniać odstępy szybciej przez każdorazowe podwajanie poprzedniego odstępu.

Kilka wzorów cieniowania uzyskanych za pomocą punktów i linii zilustrowano na rys. 3.10.

Wypełniając powierzchnię jednakową liczbą pikseli w kierunkach poziomym i pionowym można by oczekiwać, że w wyniku działania programów takich jak 3.14 otrzymamy na ekranie kwadrat. Tak jednak nie będzie w systemach, w których rozdzielczość w kierunku X różni się w istotny sposób od rozdzielczości w kierunku Y.



Rys. 3.10. Przykłady wzorów cieniowania w grafice punktowej

Na przykład może to być system, w którym poziomy odcinek złożony ze 100 pikseli ma na ekranie długość 6 cm, a odpowiedni odcinek pionowy 8 cm. Wtedy rozdzielczość w kierunku X wynosi 100/6, a w kierunku Y 100/8. Obraz złożony z 100 × 100 pikseli będzie na ekranie prostokątem trochę wyższym niż szerszym. Aby uzyskać kwadrat, można albo narysować więcej punktów w kierunku X, albo mniej punktów w kierunku Y. Załóżmy, że chcemy zmodyfikować liczbę pikseli w kierunku Y. Szukaną liczbę pikseli określa się mnożąc oryginalną długość pionowego odcinka (100) przez współczynnik równy stosunkowi rozdzielczości w kierunku Y do rozdzielczości w kierunku X (100/8 podzielone przez 100/6, czyli 3/4). W wyniku otrzymamy 75; tak więc, aby uzyskać kwadrat, musimy narysować 75 punktów w kierunku Y.

W celu określenia *współczynnika rozdzielczości* w dowolnym systemie na ekranie można narysować odcinek o długości 50 lub 100 pikseli w każdym kierunku i zmierzyć długości tych odcinków. Stosunek długości X do długości Y umożliwi skorygowanie liczby pikseli w kierunku Y, natomiast stosunek długości Y do długości X w kierunku X. Korygowanie jednego z tych kierunków w odniesieniu do

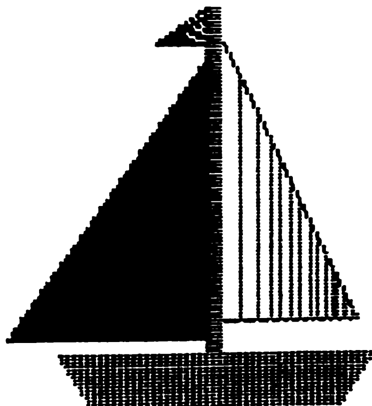
wszystkich punktów ekranu spowoduje rysowanie obrazów we właściwych proporcjach. Program 3.15 demonstruje korekcję rozdzielczości oraz sposób stosowania cieniowania i koloru przy tworzeniu obrazu (rys. 3.11).

Program 3.15. Rysunek żaglówki z cieniowaniem i korekcją rozdzielczości

```
10  'PROGRAM 3.15. Rysowanie żaglówki według szkicu na papierze milime-
20        'trowym (z korekcją dla różnych rozdzielczości X i Y)
30        'Wypełnianie i cieniowanie nieregularnych powierzchni. Punkty
40        'otrzymane na podstawie szkicu są zawarte w instrukcjach DATA.
50        'Wartości Y są mnożone przez wartość R.
60  CLEARSCREEN
70  R = 5 / 6
80  PRINT "Podaj wartości maksymalne poziomą i pionową"
90  PRINT "dla przyjętej rozdzielczości"
100 INPUT XM, YM
110 CLEARSCREEN
120 GRAPHICS
130       'Rysuj maszt łódki
140 COLOR 2,6       'Rysuj niebieski maszt na jasnoniebieskim tle
150 READ XT, YT, XB, YB
160 YT = YT * R     'Korekcja różnicy rozdzielczości
170 YB = YB * R
180 IF YT < 0 OR YT > YM OR YB < 0 OR YB > YM THEN 960
190       'Rysuj maszt o grubości 5 linii
200 FOR K = 0 TO 4
210   DRAWLINE XT+K,YT TO XB+K,YB
220 NEXT K
230       'Odczytaj współrzędne lewego boku
240       'Wypełnij dół łódki
250 COLOR 3,6
260 READ XL1, YL1, XL2, YL2
270       'Skoryguj współrzędne Y ze względu na różnicę rozdzielczości
280 YL1 = YL1 * R
290 YL2 = YL2 * R
300 IF YL1 < 0 OR YL1 > YM OR YL2 < 0 OR YL2 > YM THEN 960
310       'Znajdź nachylenie i przesunięcie prostej dla lewego boku
320 ML = (YL2 - YL1) / (XL2 - XL1)
330 BL = YL1 - ML * XL1
340       'Odczytaj współrzędne prawego boku
350 READ XR1, YR1, XR2, YR2
360       'Skoryguj współrzędne Y ze względu na różnicę rozdzielczości
370 YR1 = YR1 * R
380 YR2 = YR2 * R
390 IF YR1 < 0 OR YR1 > YM OR YR2 < 0 OR YR2 > YM THEN 960
400       'Znajdź nachylenie i przesunięcie prostej dla prawego boku
410 MR = (YR2 -YR1) / (XR2 - XR1)
420 BR = YR1 - MR * XR1
430       'Znając wartości nachyleń i przesunięć oblicz wartości X dla
440       'lewego i prawego boku i rysuj odcinki wypełniające kadłub
450 FOR Y = YL1 TO YL2
460   XL = (Y - BL) / ML
470   XR = (Y - BR) / MR
480   DRAWLINE XL,Y TO XR,Y
490 NEXT Y
500       'Rysuj duży żagiel wypełniony w sposób ciągły
510 COLOR 0,6
520 READ X1, Y1, X2, Y2
530       'Skoryguj współrzędne Y ze względu na różnicę rozdzielczości
540 Y1 = Y1 * R
550 Y2 = Y2 * R
560 IF Y1 <  0 OR Y1 > YM OR Y2 < 0 OR Y2 > YM THEN 960
570 DRAWLINE X1,Y1 TO X2,Y2
580 DRAWLINE X2,Y2 TO X1,Y2
```

Program 3.15 (cd.)

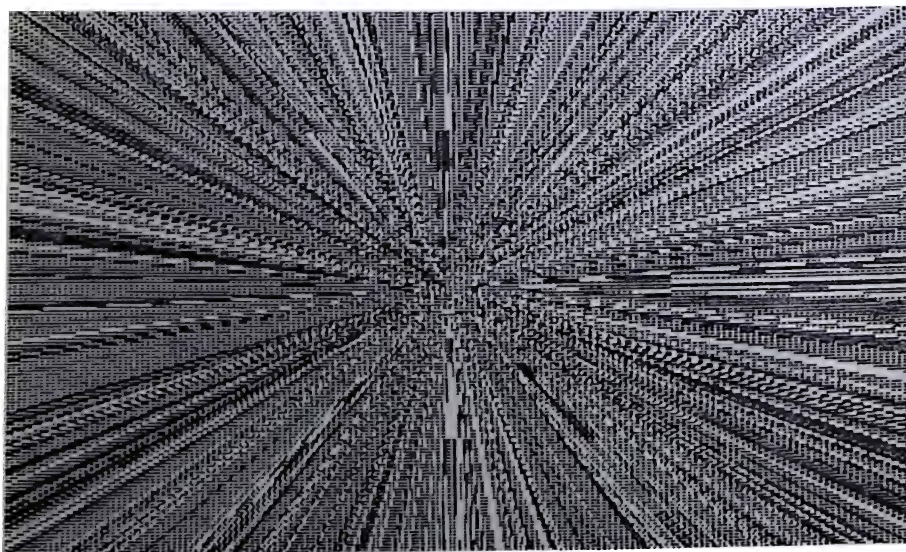
```
590 M = (Y2 - Y1) / (X2 - X1)      'Znajdź M i B dla ukośnego boku
600 B = Y1 - M * X1
610       'Oblicz punkty wzdłuż górnej krawędzi żagla
620 FOR X = X1 TO X2 STEP -1
630   Y = M * X + B
640   DRAWLINE X,Y TO X,Y2
650 NEXT X
660       'Rysuj małą flagę.
670 COLOR 3,6
680 READ X1, Y1, X2, Y2
690       'Skoryguj współrzędne Y ze względu na różnicę rozdzielczości
700 Y1 = Y1 * R
710 Y2 = Y2 * R
720 IF Y1 < 0 OR Y1 > YM OR Y2 < 0 OR Y2 > YM THEN 960
730 FOR Y = Y1 TO Y2 STEP 2
740   DRAWLINE X2,Y2 TO X1,Y
750 NEXT Y
760       'Cieniuj mały żagiel
770 C = 5
780 READ X1, Y1, X2, Y2
790       'Skoryguj współrzędne Y ze względu na różnicę rozdzielczości
800 Y1 = Y1 * R
810 Y2 = Y2 * R
820 IF Y1 < 0 OR Y1 > YM OR Y2 < 0 OR Y2 > YM THEN 960
830 DRAWLINE X1,Y1 TO X2,Y2
840 DRAWLINE X1,Y2 TO X2,Y2
850 M = (Y2 - Y1) / (X2 - X1)      'Znajdź M i B dla ukośnego boku
860 B = Y1 - M * X1
870 X = X1 + C
880       'Oblicz punkty wzdłuż górnej krawędzi żagla
890 Y = M * X + B
900 DRAWLINE X,Y TO X,Y2
910 C = C - C * 0.1
920 X = X + C
930       'Czy osiągnęliśmy prawą krawędź żagla? Jeżeli tak,to stop
940 IF X <= X2 THEN 890
950 GOTO 1020
960 PRINT "Po korekcji rozdzielczości współrzędna X poza ekranem"
970 DATA 140,10,140,110
980 DATA 95,110,105,125,190,110,180,125
990 DATA 140,25,80,105
1000 DATA 140,10,125,20
1010 DATA 145,20,185,100
1020 END
```



Rys. 3.11. Obraz uzyskany za pomocą programu 3.15 z korekcją rozdzielczości i cieniowaniem

Program 3.16. Wzory z losowo dobieranymi kolorami

```
10  'PROGRAM 3.16. Kolorowe wzory
20  CLEARSCREEN
30  PRINT "Podaj wartości maksymalne poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM, YM
60  XC = XM / 2
70  YC = YM / 2
80  CLEARSCREEN
90  GRAPHICS
100 FOR X = 0 TO XM
110   C = INT(RND(1) * 8)
120   COLOR C,0
130   DRAWLINE XC,YC TO X,0
140 NEXT X
150 FOR Y = 0 TO YM
160   C = INT(RND(1) * 8)
170   COLOR C.0
180   DRAWLINE XC,YC TO XM,Y
190 NEXT Y
200 FOR X = XM TO 0 STEP -1
210   C = INT(RND(1) * 8)
220   COLOR C,0
230   DRAWLINE XC,YC TO X,YM
240 NEXT X
250 FOR Y = YM TO 0 STEP -1
260   C = INT(RND(1) * 8)
270   COLOR C,0
280   DRAWLINE XC,YC TO 0,Y
290 NEXT Y
300 END
```



Rys. 3.12. Wzór graficzny utworzony przez program 3.16

Za pomocą programu 3.16 tworzymy na ekranie kolorowy wzór. Wynik działania programu przedstawiono na rys. 3.12. Korzystając z mechanizmu rysowania odcinków, koloru i generatora liczb losowych można tworzyć różne kolorowe wzory.

Program 3.17. Interakcyjne szkicowanie rysunków

```
10  'PROGRAM 3.17. Interakcyjne szkicowanie rysunków
20  CLEARSCREEN
30  PRINT
40  PRINT "Program umożliwia interakcyjne szkicowanie rysunków za pomo-"
50  PRINT "cą klawiatury. Program stale sygnalizuje stan oczekiwania"
60  PRINT "klawiatury na jakiś sygnał wejściowy. Program stwierdza,"
70  PRINT "który klawisz został naciśnięty i odpowiednio zmienia"
80  PRINT "wartość X lub Y"
90  PRINT
100 PRINT "Naciśnij klawisz D w celu przedłużenia linii do dołu"
110 PRINT "Nacisnij klawisz U w celu przedłużenia linii do góry"
120 PRINT "Naciśnij klawisz R w celu przedłużenia linii w prawo"
130 PRINT "Naciśnij klawisz L w celu przedłużenia linii w lewo"
140 PRINT "Naciśnij klawisz S w celu zatrzymania programu"
150 PRINT
160 PRINT "Podaj wartości maksymalne poziomą i pionową"
170 PRINT "dla przyjętej rozdzielczości"
180 INPUT XM,YM
190 PRINT
200 PRINT "Podaj współrzędne początku linii"
210 INPUT X, Y
220 CLEARSCREEN
230 GRAPHICS
240 IF X < 0 OR X > XM OR Y < 0 OR Y > YM THEN 330
250 POINTPLOT X,Y
260 A$ = INKEY$: IF A$ = "" THEN 260
270 IF A$ = "D" THEN Y = Y + 1: GOTO 240
280 IF A$ = "U" THEN Y = Y - 1: GOTO 240
290 IF A$ = "R" THEN X = X + 1: GOTO 240
300 IF A$ = "L" THEN X = X - 1: GOTO 240
310 IF A$ = "S" THEN 340
320 GOTO 240
330 PRINT "Współrzędna poza zakresem"
340 END
```

Obrazy tworzy się również *interakcyjnie*. Za pośrednictwem klawiatury można współpracować z programem w celu szkicowania (program 3.17). Program 3.17 rysuje punkty w kierunku podanym z klawiatury. Do interakcyjnego konstruowania obrazów stosuje się również inne urządzenia wejściowe, takie jak pióro świetlne, tabliczka graficzna lub inny manipulator. Urządzenia takie omawiamy dokładniej w rozdz. 11. Interakcyjną metodę konstruowania obrazu z odcinków ilustruje program 3.18.

Program 3.18. Interakcyjne konstruowanie obrazu z odcinków

```
10  'PROGRAM 3.18. Interakcyjne konstruowanie rysunku
20        'Interakcyjne rysowanie odcinków. Użytkownik może wprowadzić
30        'współrzędne początku i końca każdego odcinka,jak również
40        'żądany kolor odcinka. Program rysuje odcinek,a potem
50        'umożliwia jego zachowanie lub usunięcie.
60        '
70        '************************************************************
80  DIM X1(20), Y1(20), X2(20), Y2(20), C(20)
90  CLEARSCREEN
```

Program 3.18 (cd.)

```
100 PRINT "Podaj wartości maksymalne poziomą i pionową"
110 PRINT "dla przyjętej rozdzielczości"
120 INPUT XM, YM
130 T = 1            'T - liczba odcinków
140 CLEARSCREEN
150 GRAPHICS
160 PRINT "(Podaj -1, -1 w celu zakończenia programu)"
170        'Podaj współrzędne dla każdego odcinka
180 PRINT "Podaj pierwszy koniec odcinka";
190 INPUT X, Y
200 IF X = -1 AND Y = -1 THEN 660
210 IF X >= 0 AND X <= XM AND Y >= 0 AND Y <= YM THEN 240
220 PRINT "Pierwsza współrzędna poza zakresem. Spróbuj jeszcze raz."
230 GOTO 180
240 X1(T) = X
250 Y1(T) = Y
260 PRINT "Podaj drugi koniec odcinka";
270 INPUT X, Y
280 IF X = -1 AND Y = -1 THEN 660
290 IF X >= 0 AND X <= XM AND Y >= 0 AND Y <= YM THEN 320
300 PRINT "Druga współrzędna poza zakresem. Spróbuj jeszcze raz"
310 GOTO 260
320 X2(T) = X
330 Y2(T) = Y
340        'Wybierz kolor odcinka
350 PRINT "Podaj kolor odcinka";
360 INPUT S
370 IF S >= 0 AND S <= 15 THEN 400
380 PRINT "Kod koloru musi być w przedziale 0 - 15"
390 GOTO 350
400 C(T) = S
410 GOSUB 570                  'Rysuj obraz
420        'Czy odcinek zachować,czy odrzucić?
430 PRINT "Zachować,czy usunąć ostatni odcinek? (Z lub U)"
440 INPUT A$
450 IF A$ = "Z" OR A$ = "U" THEN 480
460 PRINT "Wprowadź tylko Z albo U"
470 GOTO 430
480 IF A$ <> "Z" THEN 530
490        'Zachowaj odcinek
500 T = T + 1                  'Można wprowadzać następny odcinek
510 GOTO 160
520        'Odrzuć odcinek
530 T = T - 1
540 GOSUB 570                  'Rysuj obraz bez ostatniego odcinka
550 T = T + 1                  'Można wprowadzać następny odcinek
560 GOTO 160
570        'Rysuj obraz
580 CLEARSCREEN
590 FOR K = 1 TO T
600   COLOR C(K),0             'Ustal kolor przedniego planu
610   DRAWLINE X1(K),Y1(K) TO X2(K),Y2(K)
620 NEXT K
630 COLOR 1,0                  'Plan przedni - kolor biały, tło - czarny
640 RETURN
650 GOSUB 570
660 END
```

ZADANIA

3.1. Korzystając z instrukcji PRINT napisz program, który będzie zerował ekran i wyświetlał wyraz HALO (lub jakikolwiek inny wyraz) napisany dużymi literami i umieszczony w środku ekranu.

3.2. Korzystając z pętli i instrukcji PRINT wyświetl następujące wzory:

```
a)  *******        (b)      @
     *****                 @@@
      ***                 @@@@@
       *                 @@@@@@@
      ***                 @@@@@
     *****                 @@@
    *******                 @
```

3.3. Narysuj jakąś figurę na papierze milimetrowym. Wypełnij figurę wybranym znakiem i wyświetl obraz na ekranie korzystając z instrukcji PRINT.

3.4. Utwórz figurę z zadania 3.3 metodą kodowania danych, tak jak to zrobiono w programie 3.4.

3.5. Napisz program tworzenia sceny z rys. 3.5 korzystając ze znaków specjalnych i instrukcji PRINT.

3.6. Narysuj obraz za pomocą instrukcji PRINT korzystając z danych zakodowanych w instrukcji DATA tak, aby można było użyć kilku różnych znaków w wierszu. Dane o każdym wierszu powinny zawierać następujące informacje: 1) kod ASCII dla każdego znaku w wierszu; 2) początkową pozycję znaku w każdym wierszu; 3) liczbę kolejnych pozycji zajmowanych przez każdy znak. Kod ASCII będzie wykorzystany przez funkcję CHR$ do wydrukowania tego znaku.

3.7. Korzystając z grafiki punktowej napisz program, który zeruje ekran i umieszcza w środku ekranu wyraz złożony z dużych liter.

3.8. Zmodyfikuj program 3.11 tak, aby rysował linię ciągłą dla wszystkich wartości współczynnika nachylenia.

3.9. Zmodyfikuj program 3.11 tak, aby rysował linię o zadanym współczynniku nachylenia i przesunięciu, biegnącą od zadanego punktu na ekranie do brzegu ekranu.

3.10. Korzystając z instrukcji rysowania punktu i równań (3.3) napisz program rysowania odcinka łączącego dwa punkty o zadanych współrzędnych.

3.11. Korzystając z instrukcji rysowania odcinka napisz program wyświetlania prostokąta, dla którego są określone: środek (XC, YC), szerokość (W) i wysokość (H). Boki prostokąta mają być równoległe do osi współrzędnych.

3.12. Zmodyfikuj program z zadania 3.11 tak, aby prostokąt był malowany określonym kolorem i była wyświetlana liczba zachodzących na siebie kolorowych prostokątów.

3.13. Zmodyfikuj program z zadania 3.11 tak, aby wypełniał prostokąt kropkami, odległość między którymi jest podana w danych wejściowych. Należy zapewnić możliwość wyświetlania określonej liczby nakładających się, zacieniowanych prostokątów.

3.14. Napisz program wyświetlania wzorów cieniowania pokazanych na rys. 3.10.

3.15. Naszkicuj rysunek na papierze milimetrowym i napisz program wyświetlania tego rysunku korzystając z instrukcji rysowania odcinka i wzorów cieniowania.

3.16. Zmodyfikuj program 3.17 tak, aby umożliwiał interakcyjne rysowanie odcinków ukośnych, poziomych i pionowych.

3.17. Zmodyfikuj program 3.17 tak, aby umożliwiał interakcyjne rysowanie odcinków o określonych długościach.

4. Proste wykresy

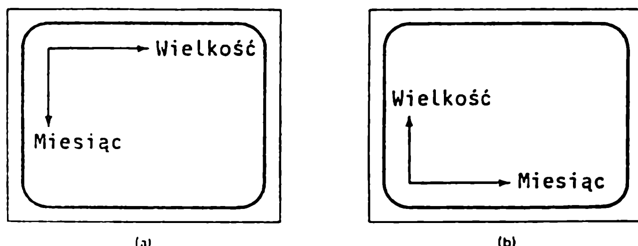
Informacje przedstawione w postaci *tabeli* są zazwyczaj trudniejsze do interpretacji niż prezentacja ich w postaci *wykresu*. Wykresy umożliwiają szybszą interpretację informacji zawartej w zbiorze liczb. W jasny sposób pokazują różne zależności, które często trudno zauważyć w zwykłym zestawieniu liczbowym. Wykresy można konstruować korzystając albo z instrukcji PRINT, albo ze specjalnych instrukcji graficznych wprowadzonych w rozdz. 3. W niektórych zastosowaniach korzystanie z instrukcji PRINT może być użyteczne, jednak instrukcje graficzne umożliwiają tworzenie dokładniejszych, bardziej przejrzystych wykresów.

4.1. Wykresy ilustrujące tendencje zmian

Elementarny wykres prezentuje ogólną tendencję lub „kształt" danych, z niewielką liczbą opisów lub bez nich. Wykres danych zestawionych na rys. 4.1 pokazuje ogólny obraz zmian sprzedaży w ciągu roku. Wykres może być tak zorientowany, żeby wielkości sprzedaży były reprezentowane w kierunku poziomym bądź pionowym. Takie

Miesiąc	Liczba sprzedanych jednostek	Miesiąc	Liczba sprzedanych jednostek
Styczeń	210	Lipiec	410
Luty	150	Sierpień	390
Marzec	99	Wrzesień	300
Kwiecień	250	Październik	651
Maj	183	Listopad	724
Czerwiec	352	Grudzień	516

Rys. 4.1. Przykładowa tablica wielkości sprzedaży



Rys. 4.2. Wielkości sprzedaży z rys. 4.1 reprezentowane w poziomie (a) lub w pionie (b)

dwa różnie zorientowane wykresy pokazano na rys. 4.2. Na rysunku 4.2a miesiące przedstawiono w pionie od góry do dołu ekranu, a wielkości sprzedaży w poziomie od lewej do prawej strony. Na rysunku 4.2b pokazano wariant, w którym miesiące podano w poziomie, od lewej do prawej strony ekranu, a wielkości sprzedaży w pionie.

Wykresy uzyskane za pomocą instrukcji PRINT

Wykres ilustrujący zestawienie danych z rys. 4.1 możemy skonstruować, drukując znaki w miejscach ekranu odpowiadających względnym wielkościom sprzedaży. Załóżmy, że chcemy przedstawić dane na ekranie o 25 wierszach i 80 znakach w wierszu. Pozycje znaków wzdłuż każdego wiersza są numerowane od 1 do 80 od lewej do prawej strony; a wiersze od 1 do 25 od góry do dołu. Poszczególnym miesiącom przyporządkujemy co drugi wiersz licząc od góry ekranu; wielkości sprzedaży będziemy reprezentowali na pozycjach od 25 do 75 każdego wiersza. Maksymalna wielkość sprzedaży będzie odpowiadała pozycji 75 w 21 wierszu od góry ekranu (dla listopada), a minimalna pozycji 25 w wierszu 5 (dla marca).

Sposób przyporządkowania pozycji w wierszu (między 25 a 75) dla poszczególnych danych określa następująca zależność:

$$\text{Pozycja w wierszu} = (\text{wartość danej} - \text{minimalna wartość danej}) * \\ * \frac{\text{zakres pozycji w wierszu}}{\text{zakres wartości danych}} + \text{najmniejsza pozycja w wierszu} \quad (4.1)$$

Dla naszych przykładowych danych zakres wartości danych wynosi 724—99 = 625, zakres pozycji w wierszu 75—25 = 50 i najmniejsza pozycja w wierszu jest 25. Stąd pozycję każdej danej znajdujemy jako

$$\text{Pozycja w wierszu} = (\text{wartość danej} - 99) * \frac{50}{625} + 25$$

lub

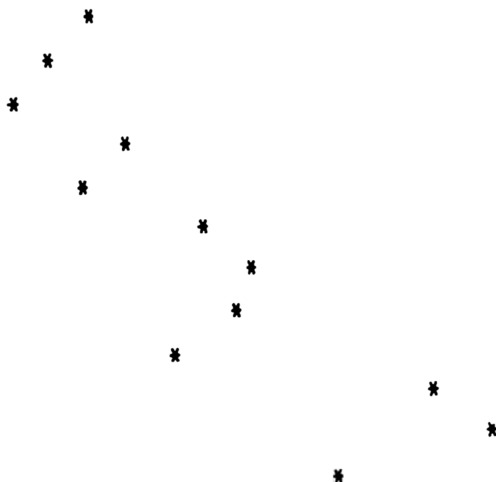
$$P = (\text{dana} - 99) * 0{,}08 + 25$$

Zależność tę wykorzystano w programie 4.1 rysującym wykres tendencji zmian pokazany na rys. 4.3. Każdą pozycję w wierszu przelicza się, a wynik zaokrągla do najbliższej liczby całkowitej korzystając z funkcji INT. Program można uogólnić dla dowolnego zbioru danych i wymiarów ekranu określając w zależności (4.1) maksymalne i minimalne wartości danych i pozycji w wierszu.

Jeżeli chcemy uzyskać wykres, w którym dane są reprezentowane w pionie (rys. 4.2b), wprowadzamy dwie zmiany w programie 4.1. Po pierwsze, dla każdej danej musimy określić właściwy wiersz, w którym będzie ona przedstawiana. Po drugie, dla każdej danej określamy pozycję w wierszu związaną z miesiącem, do któ-

Program 4.1. Poziomy wykres tendencji zmian rysowany z wykorzystaniem instrukcji PRINT

```
10  'PROGRAM 4.1. Poziomy wykres tendencji zmian uzyskany za pomocą
20       'instrukcji PRINT
30       'Wielkość sprzedaży jest reprezentowana na pozycjach 25-75,
35       'a miesiące w co drugim wierszu licząc od góry.
40  CLEARSCREEN
50  R = (75 - 25) / (724 - 99)          'R - współczynnik skali
60  FOR K = 1 TO 12
70    READ S
80    P = INT((S - 99) * R + 25 + .5)
90    PRINT TAB(P); "*"
100   PRINT
110 NEXT K
120 DATA 210,150,99,250,183,352,410,390,300,651,724,516
130 END
```



Rys. 4.3. Wykres tendencji zmian uzyskany za pomocą programu 4.1; wielkości sprzedaży z rys. 4.1 są reprezentowane w poziomie

rego ta dana się odnosi. Wartość każdej danej reprezentujemy, rysując znak w pozycji określonej przez tak ustalony wiersz i kolumnę.

Do określania wiersza i kolumny używa się specjalnych instrukcji języka Basic. W różnych systemach są to różne instrukcje. Tutaj definiujemy następującą hipotetyczną instrukcję określania wiersza i pozycji w wierszu:

> POSITION R, C — Ustawia kursor w pozycji C wiersza R. R i C mogą być stałymi lub wyrażeniami. Wartości niecałkowite są zaokrąglane.

Instrukcji POSITION używa się do określenia pozycji znaku i dotyczy ona najbliższej w programie instrukcji PRINT. Wiersze są numerowane od 1 na górze ekranu do maksymalnej wartości na dole ekranu. W dodatku A podano powiązania między tą instrukcją a instrukcjami używanymi w różnych systemach.

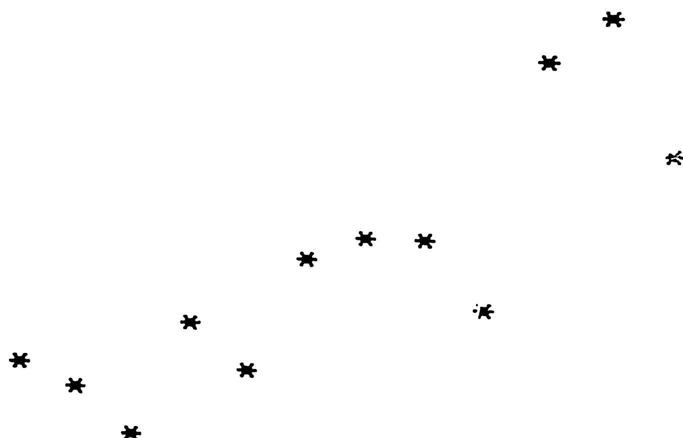
Załóżmy, że ekran ma 24 wiersze po 40 pozycji w każdym wierszu. Przyjmijmy, że każdy miesiąc będzie reprezentowany przez co trzecią kolumnę (między pozycjami 3 i 36), a wielkości sprzedaży określone przez dane z rys. 4.1 przez 20 górnych wierszy ekranu. Wtedy największa wielkość (724) będzie przedstawiona w pozycji 33 (dla listopada) górnego wiersza ekranu (wiersz nr 1). Pozycje wzdłuż wiersza ustala się jako $3 \times M$, przy czym M jest numerem miesiąca (styczeń — 1, luty — 2 itd.). Do prezentowania wielkości sprzedaży w skali od 1 do 20 korzystamy z ogólnej reguły

$$\text{Numer wiersza} = (\text{maksymalna wartość danej} - \text{wartość danej}) * \\ * \frac{\text{zakres wierszy}}{\text{zakres wartości danych}} + \text{najmniejszy numer wiersza} \quad (4.2)$$

W tym przykładzie wybraliśmy zakres wierszy równy 19 (20—1), a najmniejszy numer wiersza 1. Na rysunku 4.4 pokazano wykres poziomy otrzymany za pomocą programu 4.2.

Program 4.2. Pionowy wykres tendencji zmian rysowany z wykorzystaniem instrukcji PRINT

```
10  'PROGRAM 4.2. Pionowy wykres tendencji zmian uzyskany za pomocą
          'instrukcji PRINT
30        'Wielkość sprzedaży jest reprezentowana w wierszach 1-20,
35        'a miesiące przez co trzecią kolumnę zaczynając od 3
40        'a kończąc na 36.
50  CLEARSCREEN
60  T =(20 -1) / (724 - 99)
70  FOR M =1 TO 12
80    READ S
90    RO = INT((724 - S) * T + 1 + .5)
100   CO = M * 3
110   POSITION RO,CO
120   PRINT "*";
130 NEXT M
140 DATA 210,150,99,250,183,352,410,390,300,651,724,516
150 END
```



Rys. 4.4. Wykres tendencji zmian uzyskany za pomocą programu 4.2; wielkości sprzedaży z rys. 4.1 są reprezentowane w pionie

Program 4.2 można uogólnić tak, aby dopuszczał możliwość określania parametrów wykresu. Wtedy jako dane będzie się wprowadzać liczbę wierszy i liczbę pozycji w wierszu. Zakres wartości danych i maksymalna wartość danych zostaną określone przez program po wprowadzeniu danych.

Metoda grafiki punktowej

Korzystając ze specjalnych instrukcji graficznych do tworzenia wykresów będziemy operowali współrzędnymi ekranu, a nie wierszami i pozycjami w wierszach. Narysujmy ponownie wykres tworzony przez program 4.2, ale za pomocą instrukcji POINTPLOT. Załóżmy rozdzielczość 128 pikseli w poziomie i 48 pikseli w pionie. Wykres możemy tak umieścić, aby wykorzystać piksele w wierszach od 0 do 40 i kolumnach od 10 do 120. Miesiące będą reprezentowane przez co 10-ty piksel, zaczynając od pozycji 10. Wielkości danych reprezentowane przez wartości z zakresu 0—40 uzyskuje się z następującej zależności:

$$\text{Współrzędna Y} = (\text{maksymalna wartość danej} - \text{wartość danej}) * \\ * \frac{\text{pionowy zakres pikseli}}{\text{zakres danych}} + \text{minimalna współrzędna Y} \qquad (4.3)$$

Program 4.3 kreśli wykres tendencji zmian rozjaśniając pojedyncze piksele. Narysowanie wykresu tak, aby dane były reprezentowane w poziomie, wymaga zamiany współrzędnych X i Y miejscami i uwzględnienia wymiarów ekranu.

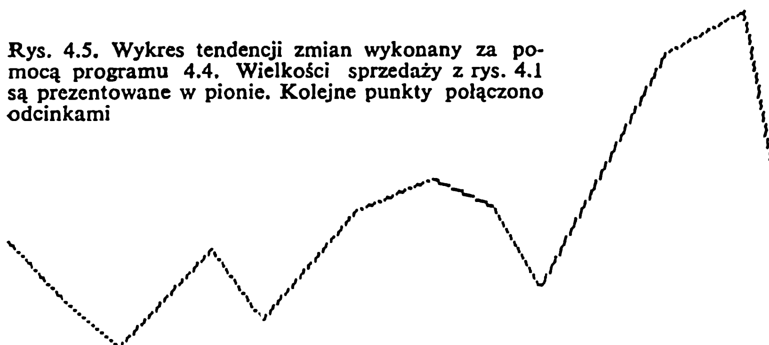
Program 4.3. Pionowy wykres tendencji zmian rysowany metodą punktową

```
10  'PROGRAM 4.3. Pionowy wykres tendencji zmian uzyskany za pomocą
20       'instrukcji POINTPLOT
30       'Wielkość sprzedaży jest reprezentowana w przedziale 0-40
35       'pikseli,a miesiące przez co 10 piksel zaczynając od 10
40       'a kończąc na 120.
50  CLEARSCREEN
60  GRAPHICS
70  R = (40 - 0) / (724 - 99)              'R - współczynnik skali
80  X = 10
90  FOR K = 1 TO 12
100   READ S
110   Y = INT((724 - S) * R + .5)
120   POINTPLOT X,Y
130   X = X + 10
140 NEXT K
150 DATA 210,150,99,250,183,352,410,390,300,651,724,516
160 END
```

Kreśląc wykresy metodą pikseli można dodać odcinki łączące punkty danych. Wykres odcinkowy przedstawiony na rys. 4.5 uzyskano za pomocą programu 4.4.

Program 4.4. Pionowy wykres tendencji zmian rysowany metodą odcinkową

```
10  'PROGRAM 4.4. Pionowy wykres tendencji zmian. Metoda rysowania
20       'odcinków
30       'Wielkość sprzedaży jest reprezentowana w przedziale 0-40
35       'pikseli,a miesiące przez co 10 piksel w przedziale 10-120.
40       'Wartości dla kolejnych miesięcy są łączone.
50  CLEARSCREEN
60  DIM X(12), Y(12)
70  R = (40 - 0) / (724 - 99)              'R - współczynnik skali
80  X1 = 10
90  FOR M = 1 TO 12
100   READ S
110   Y(K) = INT((724 - S) * R + .5)
120   X(K) = X1
130   X1 = X1 + 10
140 NEXT M
150 GRAPHICS
160 FOR K = 1 TO 11
170   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
180 NEXT K
190 DATA 210,150,99,250,183,352,410,390,300,651,724,516
200 END
```



Rys. 4.5. Wykres tendencji zmian wykonany za pomocą programu 4.4. Wielkości sprzedaży z rys. 4.1 są prezentowane w pionie. Kolejne punkty połączono odcinkami

4.2. Wykresy opisane

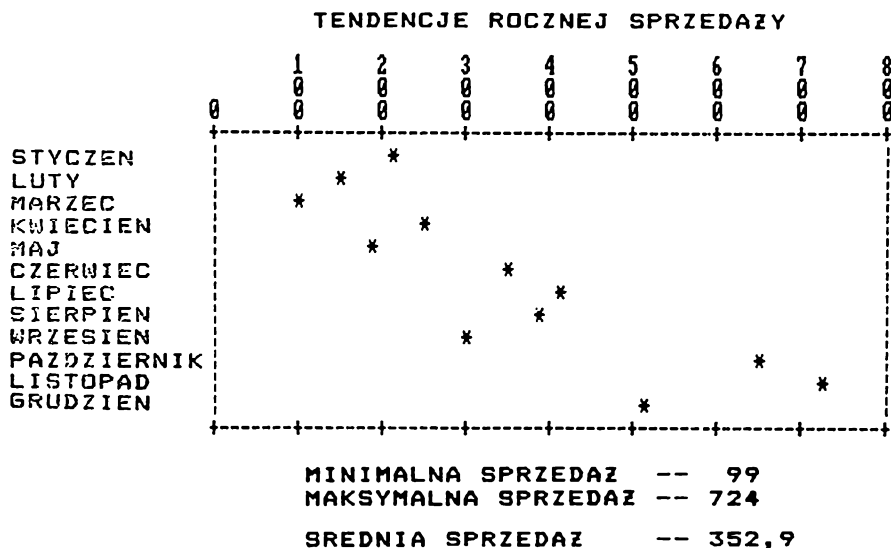
Podstawowe metody, omawiane w poprzednim punkcie, są użyteczne przy szybkim rysowaniu prostych wykresów i wyświetlaniu tendencji zmian. Jednak wykresy tendencji zmian niosą bardzo mało informacji ilościowych. Zazwyczaj interesuje nas uzyskanie z wykresu bardziej precyzyjnych informacji. Opisanie osi współrzędnych umożliwia dokładniejsze określenie zależności i interpolację danych między oznaczonymi punktami.

W przypadku wykresów opisanych konieczne jest wprowadzenie pewnych modyfikacji do zależności (4.1), (4.2) i (4.3). Zakres danych w tych zależnościach odpowiada teraz zakresowi wykresu opisanego. Jeżeli będziemy reprezentowali zestaw danych z przedziału −96 do 89 za pomocą wykresu opisanego w przedziale −100 do 100, to zakres danych wyniesie 200 (zakres wykresu). Podobnie, najmniejsza wartość danej będzie −100, a największa 100.

Osie współrzędnych X i Y można skonstruować za pomocą instrukcji PRINT i ciągu znaków plus i kreska. Przykładowy wykres opisany kreśli program 4.5. Wynik działania programu pokazano na rys. 4.6.

Program 4.5. Wykres opisany wykonany za pomocą instrukcji PRINT

```
10  'PROGRAM 4.5. Wykres opisany uzyskany za pomocą instrukcji PRINT
20      'Wielkość sprzedaży jest reprezentowana w kolumnach 12-76,
30      'miesiące w kolejnych wierszach.
40  CLEARSCREEN
50  PRINT TAB(28); "TENDENCJE ROCZNEJ SPRZEDAZY"
60  PRINT
70  PRINT TAB(12); "         1         2         3         4         5         6
    7        8"
80  PRINT TAB(12); "         0         0         0         0         0         0
    0        0"
90  PRINT TAB(12); "0        0         0         0         0         0         0
    0        0"
100 PRINT TAB(12); "+-------+-------+-------+-------+-------+-------+---
----+-------+"
110 R = (76 - 12) / (800 - 0)
120 LO = 10000
130 HI = 0
140 T = 0               'T - suma sprzedaży
150 FOR K = 1 TO 12
160   READ M$, S
170   IF S < LO THEN LO = S
180   IF S > HI THEN HI = S
190   T = T + S
200   P = INT((S-0) * R + 12 + 0.5)
210   PRINT M$; TAB(12); "|"; TAB(P); "*"; TAB(76); "|"
220 NEXT K
230 PRINT TAB(12); "+-------+-------+-------+-------+-------+-------+---
----+-------+"
240 PRINT
250 PRINT TAB(25); "MINIMALNA SPRZEDAZ  --"; USING "###"; LO
260 PRINT TAB(25); "MAKSYMALNA SPRZEDAZ --"; USING "###"; HI
270 PRINT
280 PRINT TAB(25); "SREDNIA SPRZEDAZ --"; USING "###.#"; T/12
290 DATA "STYCZEN",210,"LUTY",150,"MARZEC",99
300 DATA "KWIECIEN",250,"MAJ",183,"CZERWIEC",352
310 DATA "LIPIEC",410,"SIERPIEN",390,"WRZESIEN",300
320 DATA "PAZDZIERNIK",651,"LISTOPAD",724,"GRUDZIEN",516
330 END
```



Rys. 4.6. Wykres opisany z osiami współrzędnych wykonany za pomocą instrukcji PRINT — wynik działania programu 4.5

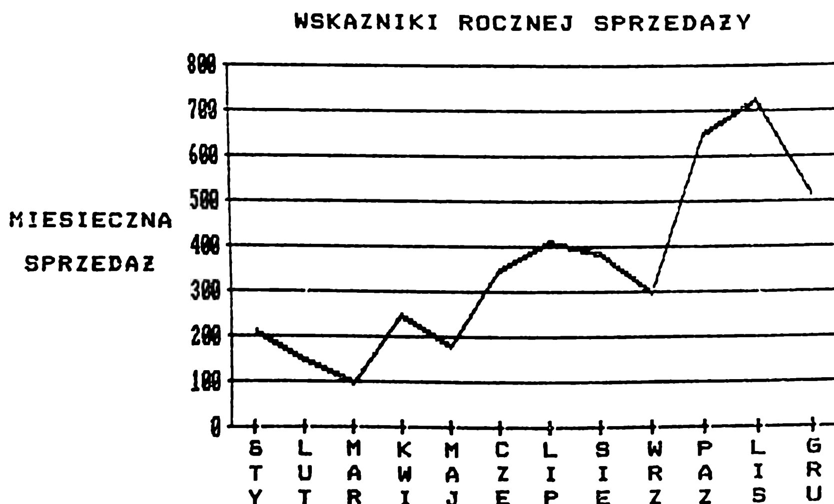
W grafice punktowej bardziej elastyczna metoda tworzenia osi współrzędnych polega na rysowaniu odcinków. Z metody tej korzysta się w programie 4.6 przy tworzeniu wykresu opisanego. Wynik działania programu zilustrowano na rys. 4.7.

Program 4.6. Wykres opisany wykonany metodą odcinkową

```
10  'PROGRAM 4.6. Wykres opisany.Metoda rysowania odcinków
20       'Założone wymiary ekranu 640 x 200 pikseli lub 25 wierszy po
30       '80 znaków. Wartości sprzedaży (z przedziału 0-800) są repre-
40       'zentowane w przedziale 27-155 pikseli,a miesiące przez
50       'co 40 piksel zaczynając od 148.
60  DIM X(12), Y(12)
70  CLEARSCREEN
80  GRAPHICS
90  PRINT
100 PRINT TAB(31); "WSKAZNIKI ROCZNEJ SPRZEDAZY"
110 DRAWLINE 128,27 TO 128,155
120 DRAWLINE 608,155 TO 608,27
130      'Rysuj znaczniki osi Y (wielkość sprzedaży) oraz linie wykresu
140 FOR Y = 27 TO 155 STEP 16
150   DRAWLINE 125,Y TO 608,Y
160 NEXT Y
170      'Opisz znaczniki
180 RO = 20          'Początek w wierszu 20
190 FOR S = 0 TO 800 STEP 100
200   POSITION RO,13
210   PRINT USING "###"; S
```

Program 4.6 (cd.)

```
220   RO = RO - 2
230 NEXT S
240        'Rysuj znaczniki dla miesięcy
250 FOR CO = 19 TO 74 STEP 5
260   POSITION 20,CO
270   PRINT "+";
280 NEXT CO
290 PRINT TAB(19);"S    L    M    K    M    C    L    S    W    P    L
G"
300 PRINT TAB(19);"T    U    A    W    A    Z    I    I    R    A    I
R"
310 PRINT TAB(19);"Y    T    R    I    J    E    P    E    Z    Z    S
U"
320 POSITION 11,3
330 PRINT "MIESIECZNA"
340 POSITION 13,3
350 PRINT "SPRZEDAZ"
360        'Rysuj linie wykresu
370 T = (155 - 27) / (800 - 0)
380 X1 = 148
390 FOR K = 1 TO 12
400   X(K) = X1
410   X1 = X1 + 40
420   READ S
430   Y(K) = INT((800 - S) * T + 27 + 0.5)
440 NEXT K
450 FOR K = 1 TO 11
460   DRAWLINE X(K),Y(K), TO X(K+1),Y(K+1)
470   'Pogrubienie linii
480   DRAWLINE X(K),Y(K)-1 TO X(K+1),Y(K+1)-1
490 NEXT K
500 DATA 210,150,99,250,183,352,410,390,300,651,724,516
510 END
```



Rys. 4.7. Wykres opisany z osiami współrzędnych wykonany metodą grafiki punktowej za pomocą programu 4.6

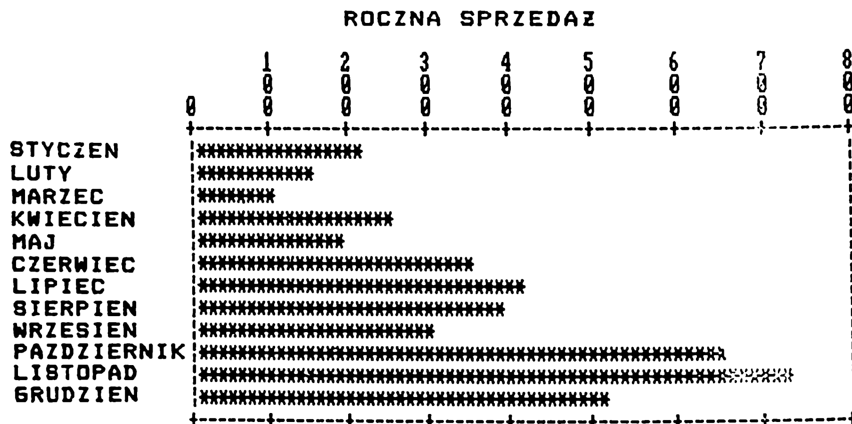
Konstruując wykresy opisane powinniśmy przestrzegać następujących zasad. Opis powinien być prosty i umieszczony we właściwym miejscu. Zbyt duża liczba opisów może zaciemnić wykres i pogorszyć jego czytelność. Ze względu na przejrzystość opisu korzystniejsze są litery i liczby duże niż małe. Jeśli to możliwe, opisy identyfikujące powinny być umieszczone na liniach lub powierzchniach, których dotyczą, a nie w oddzielnych tablicach lub legendach. Podziałka osi powinna być dobrana tak, by zapewnić łatwość interpretacji; lepiej jest wybrać podziałkę, której podstawą są wielokrotności 10 niż 8. Uwzględnienie punktu zerowego także ułatwia interpretację. Znaczniki podziałki powinny być tak rozmieszczone i opisane, by zapewnić łatwą interpolację między danymi. Linie wykresu powinny być grubsze lub bardziej intensywne niż osie współrzędnych i linie siatki. Podane zalecenia wzięto pod uwagę przy konstruowaniu wykresu z rys. 4.7.

4.3. Wykresy słupkowe — kolor i cieniowanie

Interpretację danych przedstawionych w postaci wykresu można ułatwić, prezentując wartości danych za pomocą słupków a nie punktów. Sposób ten ilustruje program 4.7 wykorzystujący instrukcje PRINT do rysowania wykresu. Na rysunku 4.8 przedstawiono wykres słupkowy (poziomy).

Program 4.7. Wykres opisany słupkowy wykonany za pomocą instrukcji PRINT

```
10  'PROGRAM 4.7. Opisany wykres słupkowy (poziomy) otrzymany za pomocą
20      'instrukcji PRINT
30      'Wartości sprzedaży są reprezentowane w kolumnach 12-76,a mie--
30      'siące w kolejnych wierszach (od góry).
40  CLEARSCREEN
50  PRINT TAB(34); "ROCZNA SPRZEDAZ"
60  PRINT
70  PRINT TAB(12); "        1       2       3       4       5       6
    7       8"
80  PRINT TAB(12); "        0       0       0       0       0       0
    0       0"
90  PRINT TAB(12); "0       0       0       0       0       0       0
    0       0"
100 PRINT TAB(12); "+-------+-------+-------+-------+-------+-------+---
----+"
110 R = (76 - 12) / (800 - 0)
120 FOR K = 1 TO 12
130   READ M$, S
140   P = INT((S - 0) * R + 12 + .5)
150   PRINT M$; TAB(12); ":";
160   FOR C = 13 TO P
170     PRINT "*";
180   NEXT C
190   PRINT TAB(76); ":"
200 NEXT K
210 PRINT TAB(12); "+-------+-------+-------+-------+-------+-------+---
----+"
220 DATA "STYCZEN",210,"LUTY",150,"MARZEC",99
230 DATA "KWIECIEN",250,"MAJ",183,"CZERWIEC",352
240 DATA "LIPIEC",410,"SIERPIEN",390,"WRZESIEN",300
250 DATA "PAZDZIERNIK",651,"LISTOPAD",724,"GRUDZIEN",516
260 END
```



Rys. 4.8. Opisany graf słupkowy uzyskany za pomocą programu 4.7, przy korzystaniu z instrukcji PRINT

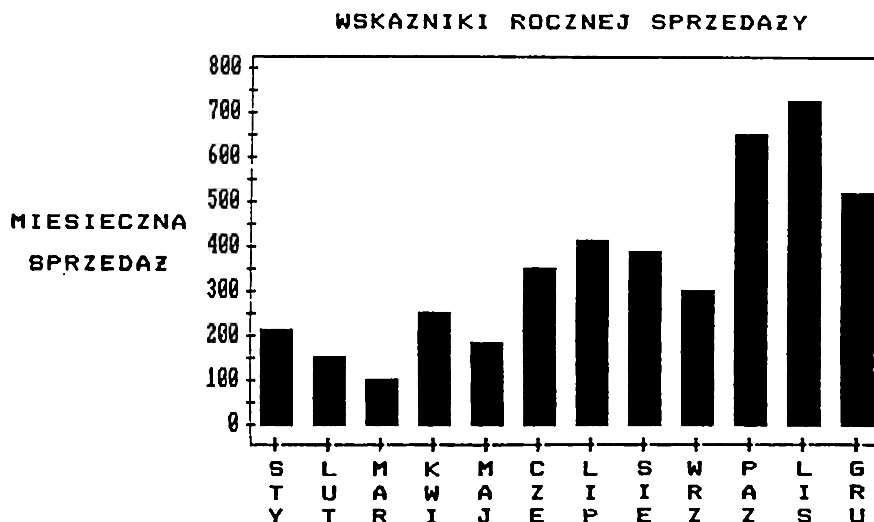
Przykładem wykorzystania grafiki punktowej do konstruowania wykresów słupkowych jest program 4.8. Punktowy wykres słupkowy pokazano na rys. 4.9. Na tym wykresie przyjęliśmy większą szerokość słupków niż odstęp między nimi. Tak się zazwyczaj postępuje, bowiem węższe słupki na ogół dają gorszy efekt wizualny.

Program 4.8. Wykres opisany słupkowy wykonany metodą odcinkową

```
10  'PROGRAM 4.8. Opisany wykres słupkowy (pionowy) wykonany metodą
20       'grafiki punktowej
30       'Założone wymiary ekranu 640 x 200 pikseli lub 25 wierszy po
40       '80 znaków. Wartości sprzedaży (z przedziału 0-800) są repre-
50       'zentowane w przedziale 27-155 pikseli.a miesiące przez co 40
55       'piksel zaczynając od 148.
60  DIM X(12), Y(12)
70  CLEARSCREEN
80  GRAPHICS
90  PRINT
100 PRINT TAB(31); "WSKAZNIKI ROCZNEJ SPRZEDAZY"
110 DRAWLINE 128,24 TO 128,163
120 DRAWLINE 128,163 TO 608,163
130 DRAWLINE 608,163 TO 608,24
140 DRAWLINE 608,24 TO 128,24
150        'Rysuj znaczniki osi Y (wielkość sprzedaży)
160 FOR Y = 27 TO 155 STEP 8
170   DRAWLINE 125,Y TO 131,Y
180 NEXT Y
190        'Opisz znaczniki
200 RO = 20          'Początek w wierszu 20
210 FOR S = 0 TO 800 STEP 100
220   POSITION RO,13
230   PRINT USING "###"; S
240   RO = RO - 2
250 NEXT S
260        'Rysuj znaczniki dla miesięcy
270 FOR CO = 19 TO 74 STEP 5          'Początek w kolumnie 19
```

Program 4.8 (cd.)

```
280   POSITION 21,CO
290   PRINT "+";
300 NEXT CO
310 PRINT TAB(19);"S    L    M    K    M    C    L    S    W    P    L
G"
320 PRINT TAB(19);"T    U    A    W    A    Z    I    I    R    A    I
R"
330 PRINT TAB(19);"Y    T    R    I    J    E    P    E    Z    Z    S
U"
340 POSITION 11,3
350 PRINT "MIESIECZNA"
360 POSITION 13,3
370 PRINT "SPRZEDAZ"
380        'Rysuj słupki wykresu
390 T = (155 - 27) / (800 - 0)
400 X1 = 136          'Pierwszy słupek zaczyna się w 136
410 FOR K = 1 TO 12
420   READ S
430   Y = INT((800 - S) * T + 27 + .5)
440   FOR X = X1 TO X1+24
450     DRAWLINE X,Y TO X,155
460   NEXT X
470   X1 = X1 + 40
480 NEXT K
490 DATA 210,150,99,250,183,352,410,390,300,651,724,516
500 END
```



Rys. 4.9. Opisany graf słupkowy uzyskany za pomocą programu 4.8; wykorzystano metodę grafiki punktowej

Większą przejrzystość i lepszą jakość wykresu słupkowego można uzyskać wykorzystując kolor. Wybór kombinacji kolorów musi być bardzo staranny; problem ten omówiono w p. 3.5. Użycie zbyt dużej liczby kolorów albo kolorów kłócących się ze sobą może pogorszyć jakość wykresu.

W wykresach kolorowych lub czarno-białych można używać różnych wzorów cieniowania. Podobnie jak w przypadku kolorów powinniśmy tak dobierać wzory, aby nie pogorszyć jakości wykresu. Należy unikać dziwacznych lub nie harmonizujących ze sobą wzorów. W przypadku cieniowania sąsiadujących powierzchni najkorzystniejsze jest stopniowanie cieniowania. Zgodnie z tym zaleceniem program 4.9 kreśli wykres z cieniowaniem pokazany na rys. 4.10.

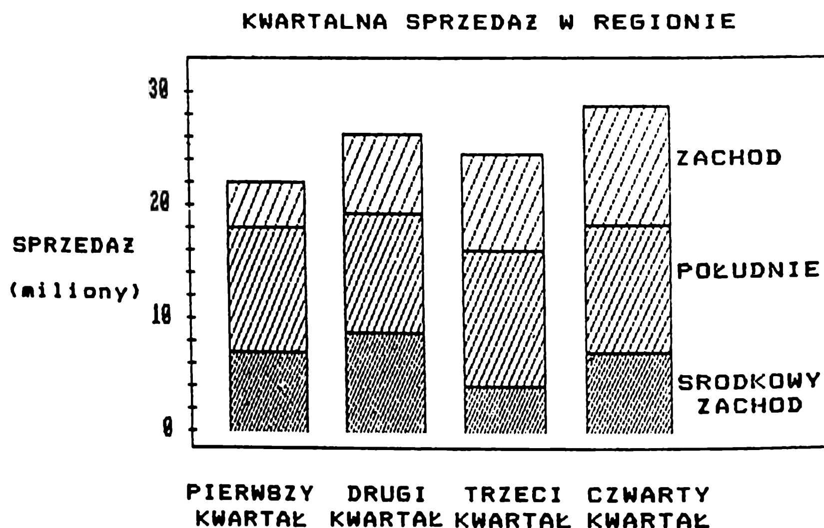
Wykresy można konstruować interakcyjnie, tak jak w przypadku rysunków. Wykres możemy opracowywać korzystając z klawiatury, pióra świetlnego lub tabliczki graficznej. Po uzyskaniu wykresu na ekranie możemy dodać osie i opis wykresu oraz zapamiętać go, aby w przyszłości wykorzystać w raportach lub prelekcjach. Wykres możemy również powiększać, zmniejszać lub dokonywać jego transformacji jednym ze sposobów dyskutowanych w części III.

Program 4.9. Wykres słupkowy cieniowany wykonany metodą grafiki punktowej

```
10  'PROGRAM 4.9. Cieniowany wykres słupkowy wykonany metodą grafiki
20       'punktowej
30       'Założone wymiary ekranu: 640 x 200 pikseli, 25 wierszy po 80
40       'znaków. Wartość sprzedaży jest reprezentowana w przedziale
50       '36-146. Szerokość paska reprezentującego kwartał wynosi
60       '64 piksele, początek w 160 kolumnie.
70       '*************************************************************
80  CLEARSCREEN
90  GRAPHICS
100 PRINT
110 PRINT TAB(28); "KWARTALNA SPRZEDAZ W REGIONIE "
120      'Rysuj granice pola wykresu"
130 DRAWLINE 128,24 TO 128,162
140 DRAWLINE 128,162 TO 608,162
150 DRAWLINE 608,162 TO 608,24
160 DRAWLINE 608,24 TO 128,24
170      'Rysuj znaczniki dla wartości sprzedaży
180 FOR Y = 36 TO 156 STEP 8
190   DRAWLINE 125,Y TO 131,Y
200 NEXT Y
210      'Opisz znaczniki
220 RO = 20
230 FOR S = 0 TO 30 STEP 10
240   POSITION RO,13
250   PRINT USING "#"; S
260   RO = RO - 5
270 NEXT S
280      'Opisz kwartały
290 POSITION 23,1
300 PRINT TAB(22); "PIERWSZY     DRUGI       TRZECI       CZWARTY
310 PRINT TAB(22); "KWARTAŁ      KWARTAŁ     KWARTAŁ      KWARTAŁ
320 POSITION 12,3
330 PRINT "SPRZEDAZ"
340 POSITION 14,2
350 PRINT "(miliony)"
```

Program 4.9 (cd.)

```
360       'Opisz regiony
370 POSITION 8,67: PRINT "ZACHOD"
380 POSITION 13,67: PRINT "POŁUDNIE"
390 POSITION 18,67: PRINT "SRODKOWY ZACHOD"
400       '************* Rysuj słupki po jednym na kwartał ***********
410 T = (156 - 36) / (30 - 0)
420 XL = 160          'Słupek dla pierwszego kwartału zaczyna się od
430 XR = XL + 64                                         'piksela 160
440 FOR Q = 1 TO 4  'Są cztery kwartały
450   YB = 156
460   'A wykorzystuje się do korygowania wartości YT (skalowana wartość
470   'sprzedaży) dla każdego regionu. Korekta jest konieczna,ponieważ
480   'cieniowane obszary reprezentujące sprzedaż kwartalną w poszcze-
490   'gólnych obszarach są umieszczone jeden nad drugim
500   A = 0
510   FOR D = 1 TO 3      'W każdym kwartale podano dane dla 3 regionów
520     READ S
530     'Zamień na miliony
540     S = S / 10 ^ 6
550     YT = INT((30 - S) * T + 36 + 0.5)
560     'Korekta YT o wartość A. A jest obszarem,który został zajęty na
570     'prezentację wielkości sprzedaży w poprzednich regionach.
580     YT = YT - A
590     DRAWLINE XL,YT TO XL,YB     'Rysowanie granicy dla tej części
600     DRAWLINE XR,YT TO XR,YB
610     DRAWLINE XL,YT TO XR,YT
620     'Wypełnij górny prawy trójkąt tej części
630     FOR X1 = XR TO XL STEP -D * 3
640       X = X1
650       Y = YT
660       POINTPLOT X,Y
670       Y = Y + 1
680       X = X - 1
690       IF Y <= YB AND X > XL THEN 660
700     NEXT X1
710     'Wypełnij dolny lewy trójkąt tej części
720     FOR Y1 = YT TO YB STEP D * 3
730       Y = Y1
740       X = XR
750       POINTPLOT X,Y
760       Y = Y + 1
770       X = X - 1
780       IF Y <= YB AND X > XL THEN 750
790     NEXT Y1
800     'Określ współczynnik korygujący dla następnego regionu
810     A = 156 - YT
820     YB = YT - 1
830   NEXT D
840   'Przejdź do słupka dla następnego kwartału
850   XL = XL + 96
860   XR = XR + 64
870 NEXT Q
880 DATA 7000000,11000000,4000000
890 DATA 8800000,10500000,7000000
900 DATA 4000000,12000000,8500000
910 DATA 7000000,11333000,10500000
920 END
```



Rys. 4.10. Wykres słupkowy z cieniowaniem otrzymany za pomocą programu 4.9

ZADANIA

4.1. Napisz program wyświetlania wykresu tendencji zmian z wartościami danych reprezentowanymi w poziomie, korzystając z instrukcji PRINT i zależności (4.1) w celu umieszczenia wykresu na ekranie. Dane wejściowe do programu zawierają najmniejszą i największą pozycję w wierszu oraz najmniejszą i największą wartość danych. Należy obliczać zakres danych, zakres pozycji w wierszu, pozycję kreślenia dla każdej danej. Dopuść dowolną liczbę danych.

4.2. Napisz program wyświetlanie wykresu tendencji zmian z wartościami danych reprezentowanymi w poziomie, korzystając z instrukcji PRINT. Korzystaj z programu określania najmniejszej i największej wartości danych w wejściowym zbiorze danych zawierającym dowolną liczbę danych. Wprowadź najmniejszy i największy numer wiersza i oblicz zakres wierszy i zakres wartości danych. Pozycje poszczególnych punktów dla danych określ zgodnie z zależnością (4.2).

4.3. Zmodyfikuj program 4.7 tak, by kreślił pionowy wykres słupkowy, korzystając z instrukcji PRINT.

4.4. Napisz program kreślenia poziomego wykresu tendencji zmian metodą punktową. Dane wejściowe zawierają najmniejszą i największą pozycję piksela. Najmniejszą i największą wartość danych określ na podstawie danych wejściowych. Oblicz: zakres wartości danych, zakres pikseli w poziomie oraz współrzędną X dla każdej wartości danej.

4.5. Zmodyfikuj program z zadania 4.4, dodając łączenie punktów reprezentujących dane przez odcinki linii prostych tak, by uzyskać linię łamaną.

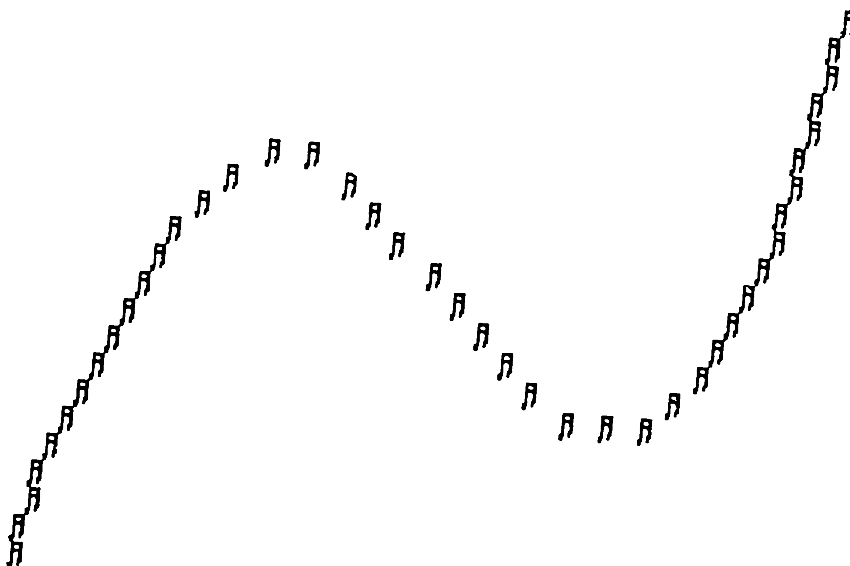
4.6. Zmodyfikuj program z zadania 4.4, dodając odcinki linii prostych (łączące punkty reprezentujące dane) i opis osi współrzędnych.

4.7. Zmodyfikuj program 4.9 tak, by kreślił poziomy wykres słupkowy. Dopuść możliwość wyboru kolorów i wzorów cieniowania jako parametrów wejściowych.

4.8. Napisz program interakcyjnego tworzenia wykresu. Punkty lub słupki powinny być wyświetlane po wprowadzeniu kolejnej danej. Na początku można podać najmniejszą i największą wartość danych oraz zakres pikseli.

5. Linie krzywe

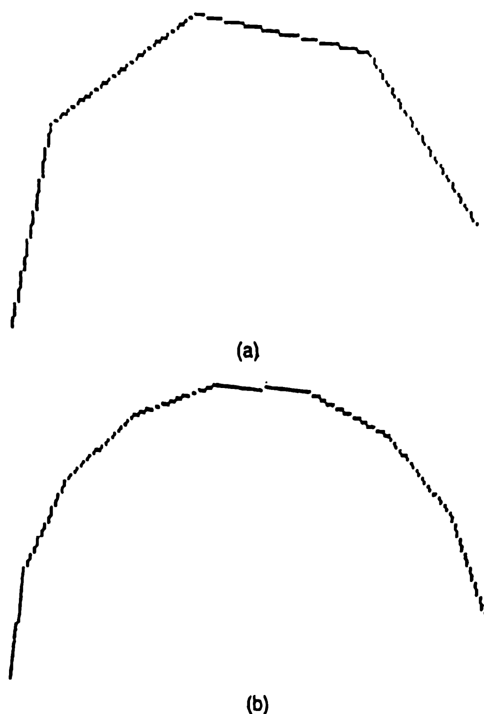
Posługując się podstawowymi metodami wprowadzonymi w poprzednich rozdziałach, można do rysunków i wykresów wykorzystać linie krzywe. Na rysunku 5.1 pokazano przybliżenie kształtu linii krzywej otrzymane za pomocą instrukcji PRINT.



Rys. 5.1. Aproksymacja ogólnego kształtu krzywej za pomocą znaków drukarskich

Pozycje poszczególnych znaków wzdłuż krzywej określamy z równania krzywej lub z wykresu krzywej na papierze milimetrowym. Ponieważ instrukcja PRINT daje ograniczone możliwości wyświetlania krzywych, skoncentrujemy nasze rozważania na metodach grafiki punktowej.

Grafika punktowa umożliwia dokładniejszą reprezentację krzywych. Krzywą możemy aproksymować za pomocą odcinków lub punktów leżących blisko siebie. Im większa jest rozdzielczość używanego systemu, tym lepszą uzyskuje się aproksymację. Na rysunku 5.2 pokazano wpływ długości odcinków na kształt krzywej. Im więcej odcinków narysuje się między końcami łuku (X1, Y1) i (X2, Y2), tym krzywa jest bardziej gładka. Jednakże im więcej mamy odcinków, tym więcej czasu potrzebuje system na utworzenie obrazu. W niektórych zastosowaniach, takich jak animacja, trzeba szybko generować obrazy. Aby zwiększyć szybkość wykonywania programu, należy wybrać mniejszą liczbę punktów i odcinków aproksymujących.



Rys. 5.2. Aproksymacja krzywej za pomocą 4 odcinków (a) i 8 odcinków (b)

5.1. Okręgi

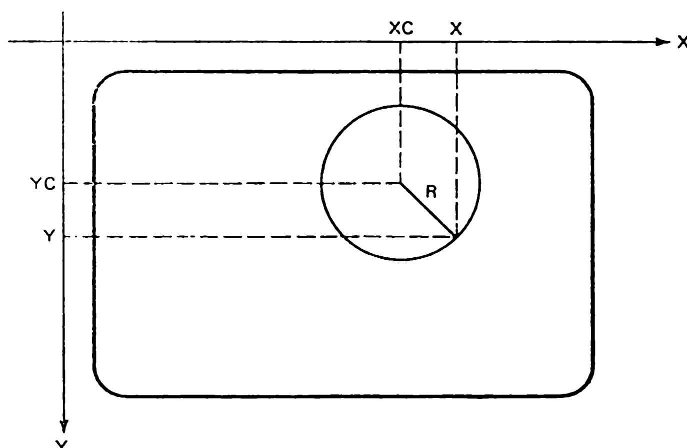
Najczęściej spotykaną krzywą jest okrąg. Krzywa ta jest bardzo użyteczna jako podstawowy element w konstruowaniu obrazów i wyświetlaniu wykresów kołowych (p. 5.4).

W celu narysowania okręgu na ekranie musimy określić jego położenie i rozmiar (rys. 5.3). Położenie okręgu określają współrzędne środka (XC, YC), a jego wielkość promień R. Potrzebujemy również reguły lub równania do wyznaczania współrzędnych pikseli wzdłuż okręgu. Równanie okręgu można przedstawić w róż-

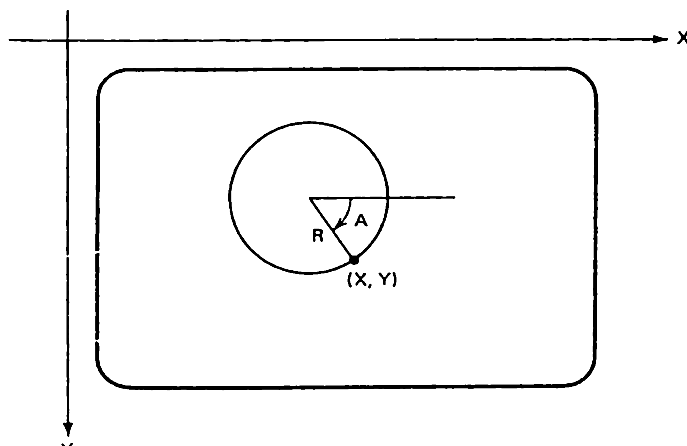
nych postaciach. Dogodnie jest określać współrzędne (X, Y) kolejnych punktów wzdłuż okręgu na podstawie wartości kąta mierzonego od linii poziomej w kierunku zgodnym z ruchem wskazówek zegara (rys. 5.4). Korzystając z tej metody możemy obliczać wartości X i Y na podstawie znajomości XC, YC, R i A jako:

$$X = XC + R * COS(A)$$
$$Y = YC + R * SIN(A) \qquad (5.1)$$

Kąt A jest mierzony w radianach; jego wartości zmieniają się przy pełnym obrocie od 0 do 6,283125 (2*PI) względem linii poziomej. Ten zakres wartości odpowiada zakresowi 0–360°.



Rys. 5.3. Koło o promieniu R i współrzędnych środka (XC, YC)



Rys. 5.4. Współrzędne punktów (X, Y) wzdłuż okręgu określone na podstawie promienia R i wartości kąta A mierzonego względem poziomu zgodnie z ruchem wskazówek zegara

Metoda rysowania za pomocą odcinków

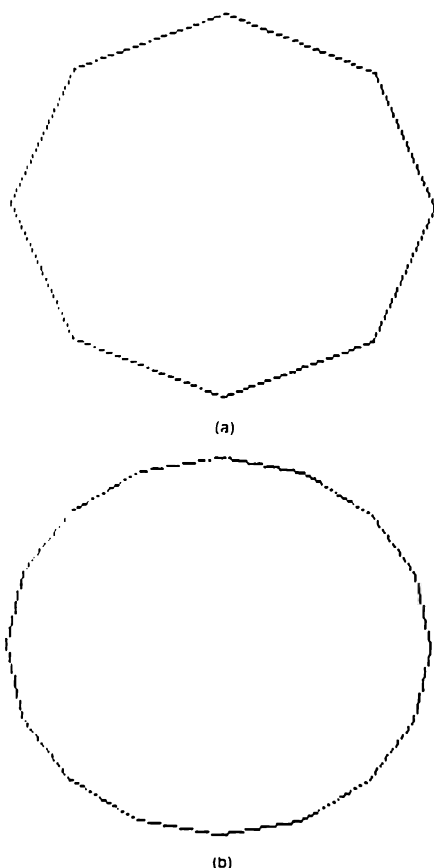
Korzystając z zależności (5.1) można napisać program wyświetlania okręgu aproksymowanego odcinkami. Program 5.1 wymaga podania parametrów XC, YC, R i liczby rysowanych punktów. Wynik działania programu pokazano na rys. 5.5.

Użycie mniejszej liczby odcinków wzdłuż okręgu przyśpieszy proces rysowania. Wtedy jednak poszczególne odcinki staną się bardziej widoczne i otrzymamy obraz bardziej przypominający wielokąt niż okrąg (rys. 5.5a).

W przypadku systemu o różnych rozdzielczościach w kierunkach X i Y możemy, rysując okrąg, otrzymać elipsę. Zgodnie z tym, co powiedziano w p. 3.5, różnice rozdzielczości możemy skompensować korygując wartości Y. Chcąc to uwzględnić w naszym programie, składnik R * SIN(A) mnożymy przez stosunek rozdzielczości Y do rozdzielczości X.

Program 5.1. Rysowanie okręgu metodą kreślenia odcinków i przyrostów kątowych

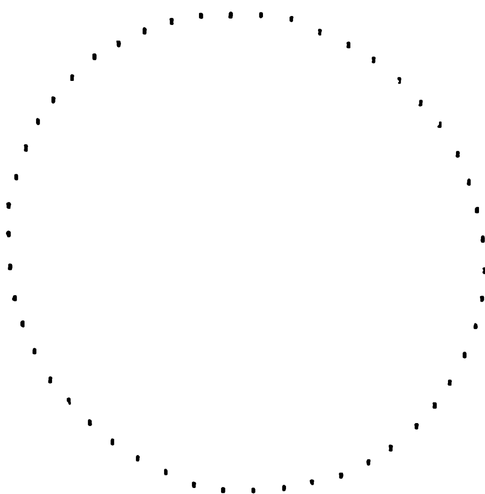
```
10  'PROGRAM 5.1. Kreślenie okręgu metodą rysowania odcinków
20  CLEARSCREEN
30  PRINT "Podaj wartości maksymalne poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM, YM
60  PRINT
70  PRINT "Podaj współrzędne środka okręgu"
80  INPUT XC, YC
90  IF XC < 0 OR XC > XM OR YC < 0 OR YC > YM THEN 350
100 PRINT
110 PRINT "Podaj promień koła"
120 INPUT R
130 IF R < 0 THEN 350
140 IF XC + R > XM OR XC - R < 0 OR YC + R > YM OR YC - R < 0 THEN 350
150 PRINT
160 PRINT "Podaj liczbę rysowanych punktów"
170 INPUT N
180       'RE = Kąt w radianach równoważny kątowi 360 stopni
190 RE = 360 * 3.141593 / 180
200 DA = RE / N          'Podziel okrąg na N części
210 CLEARSCREEN
220 GRAPHICS
230 X1 = XC + R
240 Y1 = YC
250 FOR A = DA TO RE STEP DA
260   X2 = XC + R * COS(A)
270   Y2 = YC + R * SIN(A)
280   DRAWLINE X1,Y1 TO X2,Y2
290   'Zapamiętaj następny punkt końcowy
300   X1 = X2
310   Y1 = Y2
320 NEXT A
330 DRAWLINE X1,Y1 TO XC+R,YC
340 GOTO 390
350 PRINT "Współrzędna poza zakresem. Napisz: K(koniec) lub P(powtórz)"
360 INPUT C$
370 IF C$ = "K" THEN 390
380 IF C$ = "P" THEN 70
390 END
```



Rys. 5.5. Koło aproksymowane za pomocą 8 odcinków (a) i 16 odcinków (b) — wynik działania programu 5.1

Metody rysowania za pomocą punktów

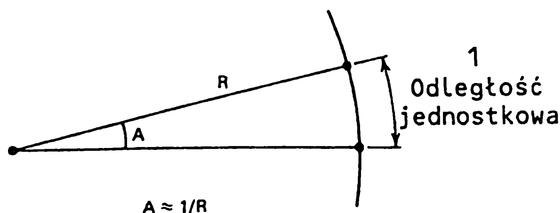
Szybkość rysowania okręgu można zwiększyć, wyświetlając należące do niego punkty i pomijając odcinki łączące te punkty. Oszczędza się w ten sposób czas przeznaczony na rysowanie odcinków. Jeżeli kolejno rysowane punkty nie są umieszczone zbyt daleko od siebie, to dzięki własnościom oka uzyska się efekt ciągłości (rys. 5.6). Im więcej narysujemy punktów, tym lepszą uzyskamy aproksymację. Najlepszą aproksymację okręgu otrzymamy wtedy, kiedy punkty będą leżały tak blisko siebie, jak tylko jesteśmy w stanie narysować. W takim przypadku piksele są rysowane w punktach siatki najbliższych pożądanym punktom należącym do okręgu. Odległość między sąsiednimi punktami siatki jest równa odległości jednostkowej w obu kierunkach poziomym i pionowym. Stąd odległość kątową (w radianach) między dwoma punktami siatki na okręgu można aproksymować jako od-



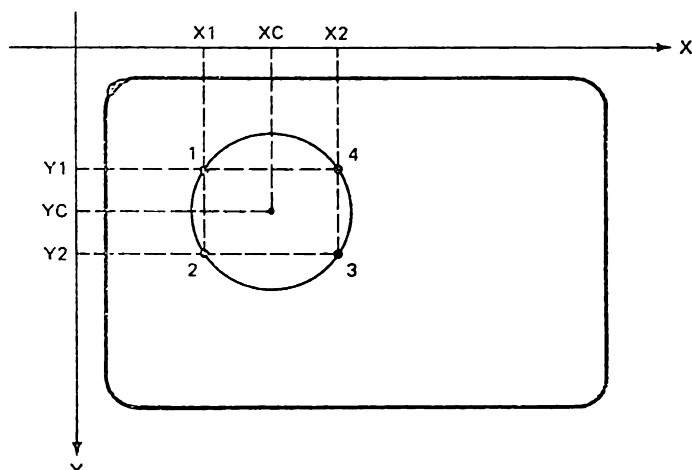
Rys. 5.6. Koło rysowane za pomocą punktów

wrotność promienia koła (rys. 5.7). W większości przypadków taka aproksymacja jest wystarczająca. Przy mniejszym kroku kątowym unikniemy przerw w linii reprezentującej okrąg; podwoimy jednak liczbę obliczeń i niektóre punkty będą rysowane podwójnie.

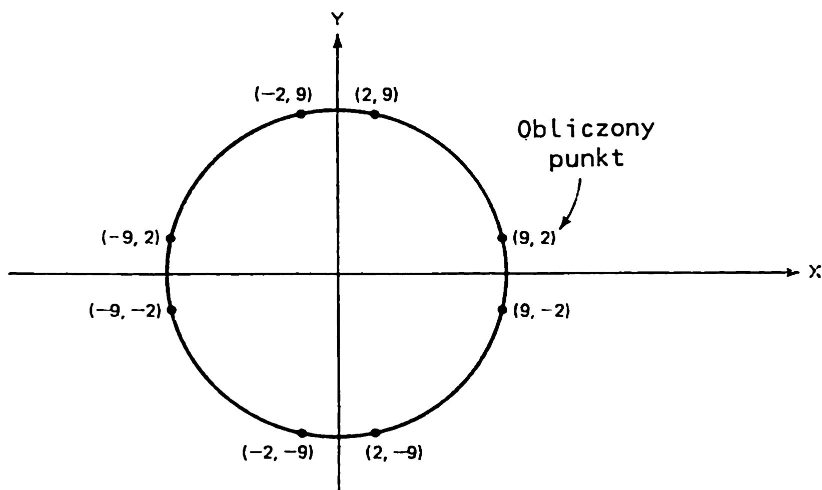
Inny sposób skrócenia czasu wyświetlania okręgu polega na wykorzystaniu własności symetrii okręgu. Nie musimy osobno liczyć każdego punktu na krzywej. Górna połowa ma taki sam kształt jak dolna, a lewa połowa ma taki sam kształt jak prawa. Oznacza to, że każda wartość X na okręgu odpowiada dwóm wartościom Y, a każda wartość Y odpowiada dwóm wartościom X (rys. 5.8). Rozwijając tę ideę dalej, możemy otrzymać następne cztery punkty okręgu zamieniając wszystkie współrzędne X i Y. To znaczy, jeżeli punkt (X, Y) leży na okręgu, to również punkt (Y, X) leży na okręgu. Musimy wobec tego wyznaczyć tylko punkty należące do jednej ósmej okręgu (segment 45°). Znając te punkty, można wyznaczyć wszystkie pozostałe punkty okręgu. Na rysunku 5.9 pokazano osiem punktów okręgu, które można narysować obliczając współrzędne tylko jednego punktu (9, 2). W tym przypadku środek koła znajduje się w środku układu współrzędnych. Dla koła o środku w punk-



Rys. 5.7. Zależność między odległością kątową A (w radianach) dwóch punktów narysowanych w odległości jednostkowej od siebie a promieniem R koła przechodzącego przez te dwa punkty



Rys. 5.8. Symetria okręgu – punkty 1 i 2 mają taką samą wartość współrzędnej X; punkty 1 i 4 mają taką samą wartość współrzędnej Y



Rys. 5.9. Jeżeli piksel o obliczonych współrzędnych (9, 2) leży na okręgu, to można narysować 8 symetrycznych punktów

cie (XC, YC) do wszystkich wartości współrzędnych X dodajemy XC i do wszystkich wartości współrzędnych Y dodajemy YC. Przesuwamy w ten sposób okrąg ze środka układu współrzędnych do pożądanej pozycji. Problem przesuwania wyświetlanych obrazów z jednej pozycji do drugiej przedyskutujemy dokładniej w rozdz. 6.

Program 5.2. Rysowanie okręgu metodą kreślenia punktów; wykorzystano przyrosty kątowe

```
10  'PROGRAM 5.2. Kreślenie okręgu metodą rysowania punktów
20        'Wyznacza się punkty należące do okręgu w przedziale 0-45
30        'stopni. Pozostałe punkty są rysowane jako punkty symetryczne.
40  CLEARSCCREEN
50  PRINT "Podaj wartości maksymalne poziomą i pionową"
60  PRINT "dla przyjętej rozdzielczości"
70  INPUT XM, YM
80  PRINT
90  PRINT "Podaj współrzędne środka okręgu"
100 INPUT XC, YC
110 IF XC < 0 OR XC > XM OR YC < 0 OR YC > YM THEN 440
120 PRINT
130 PRINT "Podaj długość promienia okręgu"
140 INPUT R
150 IF R < 0 THEN 440
160 IF XC + R > XM OR XC - R < 0 OR YC + R > YM OR YC - R < 0 THEN 440
170 CLEARSCREEN
180 GRAPHICS
190       'RA jest współczynnikiem korekcji rozdzielczości
200 RA = 5/6
210 POINTPLOT XC+R,YC
220 POINTPLOT XC-R,YC
230 POINTPLOT XC,YC+R*RA
240 POINTPLOT XC,YC-R*RA
250 DA = 1 / R
260       'RE jest kątem w radianach równoważnym kątowi 45 stopni
270 RE = 45 * 3.141593 / 180
280 FOR A = DA TO RE STEP DA
290   DX = R * COS(A)
300   DY = R * SIN(A)
310   GOSUB 340
320 NEXT A
330 GOTO 480
340       'Rysuj wsystkie symetryczne punkty
350 POINTPLOT XC+DX,YC+DY*RA
360 POINTPLOT XC-DX,YC+DY*RA
370 POINTPLOT XC+DX,YC-DY*RA
380 POINTPLOT XC-DX,YC-DY*RA
390 POINTPLOT XC+DY,YC+DX*RA
400 POINTPLOT XC-DY,YC+DX*RA
410 POINTPLOT XC+DY,YC-DX*RA
420 POINTPLOT XC-DY,YC-DX*RA
430 RETURN
440 PRINT "Współrzędna poza zakresem. Napisz: K(koniec), P(powtórz)"
450 INPUT C$
460 IF C$ = "K" THEN 480
470 IF C$ = "P" THEN 90
480 END
```

W programie 5.2 wykorzystano metodę rysowania koła, zgodnie z którą oblicza się tylko punkty leżące na okręgu w przedziale od 0 do 45°. Pozostałe punkty otrzymuje się na zasadzie symetrii. Moglibyśmy wyeliminować cztery instrukcje rysowania punktów na początku programu 5.2, rozpoczynając pętlę z wartością A = 0. Dałoby to w efekcie zmniejszenie liczby instrukcji w programie, ale każdy z czterech początkowych punktów byłby rysowany dwukrotnie. Może się zdarzyć, że niektóre punkty będą rysowane dwukrotnie. Występuje to dla większych wartości R i jest wynikiem procesu zaokrąglania w instrukcji POINTPLOT. Program 5.2

będzie pozostawiał pewne małe przerwy w liniach reprezentujących okręgi o niedużych wartościach R. Wynika to z zaokrąglania wartości współrzędnych.

Dalsze zwiększenie szybkości programów rysowania okręgu możemy uzyskać, obliczając współrzędne rysowanych punktów tak, by nie korzystać z funkcji SIN i COS. Sposób realizacji tego polega na obliczaniu pozycji pikseli na obwodzie na podstawie współrzędnych wcześniej wyznaczonych pikseli. Jeżeli obliczone współrzędne (X1, Y1) wyznaczają punkt leżący na okręgu, to następny punkt (X2, Y2) leżący na okręgu otrzymuje się jako:

$$\begin{aligned} X2 &= XC + (X1 - XC) * CA + (Y1 - YC) * SA \\ Y2 &= YC + (Y1 - YC) * CA - (X1 - XC) * SA \end{aligned} \tag{5.2}$$

W równaniu tym CA i SA są stałymi wyznaczonymi dla ustalonego kroku kątowego DA jako

$$CA = COS(DA) \qquad SA = SIN(DA) \tag{5.3}$$

Zależności (5.2) można wykorzystać w programach 5.1 lub 5.2 do określania współrzędnych pikseli dla punktów obwodu bez konieczności obliczania w każdym kroku funkcji SIN i COS. Punkt początkowy do kreślenia okręgu na podstawie zależności (5.2) ma współrzędne X1 = XC, Y1 = YC + R. Obliczenia kończy się, gdy X2 − XC ⩾ Y2 − YC. Korzystając z tej metody możemy narysować osiem punktów na okręgu po wyznaczeniu współrzędnych jednego punktu z zależności (5.2). Korektę różnicy rozdzielczości uzyskujemy mnożąc w zależności określającej wyraz (Y1 − YC) przez stosunek rozdzielczości X do Y oraz mnożąc w zależności określającej Y2 wyraz (X1 − XC) przez stosunek rozdzielczości Y do X.

Wreszcie możemy wyeliminować zaokrąglanie konieczne w operacji POINTPLOT. Uzyskujemy to, wykorzystując kroki jednostkowe w jednym z kierunków osi współrzędnych zamiast obliczeń kątowych. W tym przypadku w obliczeniach nie korzystamy z kątów i nie ma potrzeby zaokrąglania. Takie metody są trudniejsze do sformułowania. Wykorzystuje się w nich pomysł polegający na tym, że każdorazowo poruszamy się w kierunku osi X (lub osi Y) przesuwając się o odległość jednostkową i znajdując punkt siatki położony najbliżej okręgu. Przykładem takiego podejścia jest program 5.3. Obliczenia zaczynamy od piksela o współrzędnych (XC, YC + R). W każdym kolejnym kroku dodajemy jeden do współrzędnej X, a następnie poszukujemy współrzędnej Y punktu siatki położonego najbliżej okręgu. Tą szukaną wartością Y będzie albo bieżąca współrzędna Y, albo wartość mniejsza o jeden od wartości bieżącej. Wyboru jednego z tych punktów dokonujemy, korzystając z wartości parametru P. Parametr ten umożliwia oszacowanie, który z punktów leży bliżej okręgu. Jeżeli P ⩾ 0, odejmujemy jedynkę od wartości Y; w przeciwnym przypadku zachowujemy wartość Y. Szybkość wykonania programu 5.3 można nieco zwiększyć prowadząc wszystkie obliczenia w arytmetyce liczb całkowitych, jeżeli taka opcja jest dostępna w systemie. Korektę różnicy rozdzielczości uzyskujemy

mnożąc wartości Y, obliczone w programie 5.3, przez stosunek rozdzielczości Y i X (przed wejściem do podprogramu rysowania punktów).

Program 5.3. Rysowanie okręgu metodą rysowania punktów. Metoda jednostkowych przyrostów współrzędnej X

```
10  'PROGRAM 5.3.Rysowanie okręgu
20       'Metoda obliczania współrzędnej Y dla kolejnych wartości
30       'współrzędnej X.
40  CLEARSCREEN
50  PRINT "Podaj wartości maksymalne poziomą i pionową"
60  PRINT "dla przyjętej rozdzielczości"
70  INPUT XM, YM
80  PRINT
90  PRINT "Podaj współrzędne środka okręgu"
100 INPUT XC, YC
110 IF XC < 0 OR XC > XM OR YC < 0 OR YC > YM THEN 530
120 PRINT
130 PRINT "Podaj długość promienia okręgu"
140 INPUT R
150 IF R < 0 THEN 530
160 IF XC + R > XM OR XC - R < 0 OR YC + R > YM OR YC - R < 0 THEN 530
170 CLEARSCREEN
180 GRAPHICS
190       'RA jest współczynnikiem korekcji rozdzielczości
200 RA = 5/6
210 X = 0
220 Y = R
230       'Wartość P wskazuje,która
240       'wartość Y jest położona bliżej okręgu
250 P = 3 - 2 * R
260 GOSUB 370
270 IF P >= 0 THEN 300
280 P = P + 4 * X + 6
290 GOTO 320
300 P = P + 4 * (X - Y) + 10
310 Y = Y - 1
320 X = X + 1
330 GOSUB 430
340 IF X < Y THEN 260
350 IF X = Y THEN GOSUB 370
360 GOTO 570
370       'Rysuj 4 punkty
380 POINTPLOT XC+X,YC+Y*RA
390 POINTPLOT XC-X,YC-Y*RA
400 POINTPLOT XC+Y,YC-X*RA
410 POINTPLOT XC-Y,YC+X*RA
420 RETURN
430       'Rysuj wszystkie symetryczne punkty
440 POINTPLOT XC+X,YC+Y*RA
450 POINTPLOT XC-X,YC+Y*RA
460 POINTPLOT XC+X,YC-Y*RA
470 POINTPLOT XC-X,YC-Y*RA
480 POINTPLOT XC+Y,YC+X*RA
490 POINTPLOT XC+Y,YC-X*RA
500 POINTPLOT XC-Y,YC+X*RA
510 PONITPLOT XC-Y,YC-X*RA
520 RETURN
530 PRINT "Współrzędna poza zakresem.Napisz: K(koniec), P(powtórz)"
540 INPUT C$
550 IF C$ = "K" THEN 570
560 IF C$ = "P" THEN 90
570 END
```

W niektórych systemach mikrokomputerowych jest dostępna specjalna instrukcja graficzna rysowania okręgu. Wprowadźmy taką instrukcję do naszego zbioru hipotetycznych instrukcji graficznych

> CIRCLEPLOT XC, YC, R — Rysuje okrąg o środku (XC, YC) i promieniu R. Parametry XC, YC i R mogą być stałymi numerycznymi lub wyrażeniami. Jeżeli parametry te nie są liczbami całkowitymi, podlegają zaokrąglaniu.

Instrukcja ta będzie używana we wszystkich następnych programach, w których wystąpi konieczność rysowania okręgów. Zakładamy, że instrukcja CIRCLEPLOT rysuje okrąg zgodnie z metodą stosowaną w programie 5.3. W dodatku A podano związki z podobnymi instrukcjami dostępnymi w różnych systemach. W tych systemach, w których nie ma instrukcji rysowania okręgu, program 5.3 można wprowadzić jako podprogram w Basicu.

5.2. Inne krzywe

Chociaż okrąg jest najczęściej spotykaną krzywą, w grafice znajduje zastosowanie również wiele innych krzywych. Krzywe takie wyświetla się metodami podobnymi do tych, z których korzystaliśmy w przypadku okręgów. Piksele krzywej (lub pozycje znaków) można wyznaczyć z równań krzywej. W przypadku różnic rozdzielczości dokonuje się odpowiednich korekcji. Piksele rysuje się w odpowiednich miejscach ekranu i łączy odcinkami. W przypadku niektórych krzywych można korzystać z własności symetrii lub innych własności w celu zmniejszenia ilości obliczeń.

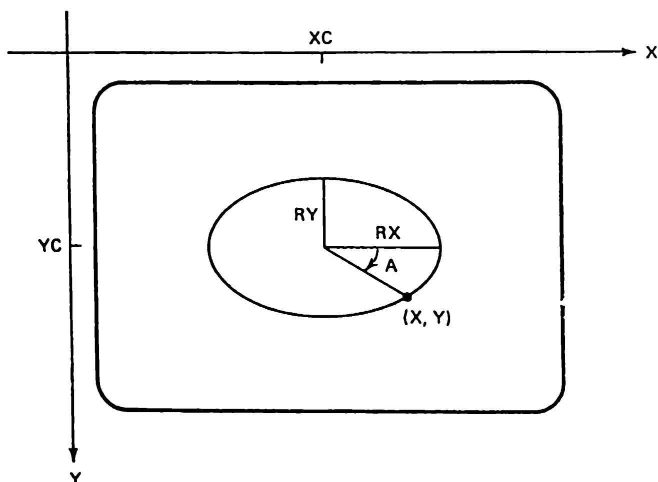
Krzywe omawiane w tym punkcie stosuje się w modelowaniu graficznym w wymienianych obszarach zastosowań lub w niektórych przypadkach do aproksymacji linii przechodzących przez punkty określone w tablicach. Metody aproksymacji krzywych (takie jak metoda najmniejszych kwadratów) umożliwiają rysowanie gładkich krzywych reprezentujących stablicowane dane.

Elipsy

Elipsę można traktować jako odmianę okręgu, chociaż, ściśle biorąc, okrąg jest specjalnym przypadkiem elipsy. Jeżeli ściśniemy okrąg w jednym kierunku (np. X), otrzymamy elipsę. Równania elipsy można zapisać w postaci:

$$\begin{aligned} X &= XC + RX * COS(A) \\ Y &= YC + RY * SIN(A) \end{aligned} \qquad (5.4)$$

W równaniach tych A jest kątem mierzonym w radianach, liczonym od poziomu w kierunku zgodnym z ruchem wskazówek zegara (rys. 5.10). Jeżeli RX > RY, oś



Rys. 5.10. Elipsa rysowana na podstawie zależności (5.4), przy czym RX > RY; (XC, YC) – współrzędne środka

elipsy w kierunku X jest dłuższa niż oś w kierunku Y. Jeżei RY > RX, oś elipsy w kierunku Y jest dłuższa niż oś w kierunku X. W przypladku gdy RX = RY, otrzymujemy okrąg.

Elipsy są użyteczne w wielu zastosowaniach modelowania graficznego. Elipsami są orbity satelitów. Kształty eliptyczne mają niektóre części maszyn. Podstawa walca widziana pod pewnym kątem wygląda jak elipsa. Możliwość bezpośredniego wyświetlania elipsy zwiększa elastyczność tworzenia obrazów w różnych zastosowaniach graficznych.

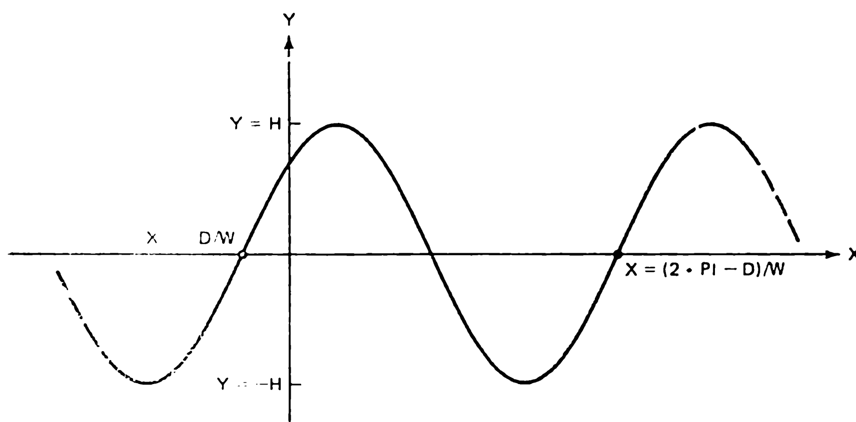
Programy rysowania okręgów podane w p. 5.1 można tak modyfikować, by kreśliły okręgi bądź elipsy. Możemy to uzyskać, zastępując równania (5.1) bardziej ogólnymi równaniami (5.4). W niektórych mikrokomputerach dostępna jest, jako opcja, możliwość rysowania elipsy za pomocą instrukcji rysowania okręgu.

Sinusoidy

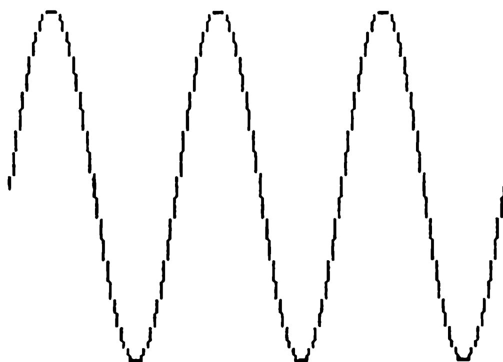
Ogólne równanie sinusoidy możemy zapisać jako

$$Y = H * SIN(W * X + D) \tag{5.5}$$

Wykres sinusoidy w konwencjonalnym układzie współrzędnych przedstawiono na rys. 5.11. Częstotliwość W wyznacza liczbę oscylacji (okresów) krzywej w określonym przedziale X. Parametr D określa przesunięcie krzywej w prawo lub w lewo. Jeżeli D ma wartość równą zeru, otrzymujemy standardową sinusoidę; dla D =



Rys. 5.11. Standardowa sinusoida o równaniu (5.5). Jeden okres krzywej narysowano między wartościami X = −D/W i X = (2 * PI − D)/W. Wartości Y oscylują między maksymalną wartością H a minimalną wartością −H



Rys. 5.12. Trzy okresy sinusoidy narysowane za pomocą programu 5.4

= PI/2 otrzymujemy cosinusoidę. Na rysunku 5.12 przedstawiono trzy okresy sinusoidy dla następujących wartości parametrów: amplituda H = 50, W = 2 * PI/50, D = 0 w przedziale od X = 0 do X = 150. Krzywą tę uzyskano za pomocą programu 5.4.

Program 5.4 wykonuje kroki jednostkowe w kierunku X, oblicza wartości Y i otrzymane punkty łączy odcinkami. Program ten nie korzysta z własności symetrii. Można jednak każdą „ćwiartkę" sinusoidy otrzymać na podstawie znajomości punktów z „pierwszej ćwiartki" ograniczonej przez wartości X = −D/W i X = = (PI/2 − D)/W. To znaczy, że jeśli (X, Y) są współrzędnymi punktu z pierwszej

Program 5.4. Rysowanie sinusoidy

```
10  'PROGRAM 5.4. Rysowanie sinusoidy
20  CLEARSCREEN
30  PRINT "Program rysuje sinusoidę o równaniu"
40  PRINT
50  PRINT "     Y = H * SIN (W * X + D)      "
60  PRINT
70  PRINT "Podaj maksymalną wartość pionową dla przyjętej"
75  PRINT "rozdzielczości"
80  INPUT YM
90  M = YM / 2
100 PRINT "Podaj amplitudę krzywej H, częstotliwość W, przesunięcie D"
110 PRINT "H nie może być większe niż połowa wysokości ekranu YM"
120 PRINT
130 INPUT H, W, D
140 IF H > M THEN 100
150 PRINT "Podaj minimalną i maksymalną wartość X"
160 INPUT XL, XR
170 CLEARSCREEN
180 GRAPHICS
190 Y1 = H * SIN(W * XL + D)
200 IF Y1 >= 0 THEN Y1 = M - Y1
210 IF Y1 < 0 THEN Y1 = M + ABS(Y1)
220 X1 = XL
230 FOR X2 = XL TO XR
240   Y2 = H * SIN(W * X2 + D)
250   IF Y2 >= 0 THEN Y2 = M - Y2
260   IF Y2 < 0 THEN Y2 = M + ABS(Y2)
270   DRAWLINE X1,Y1 TO X2,Y2
280   'Zapamiętaj nowy punkt
290   X1 = X2
300   Y1 = Y2
310 NEXT X2
320 END
```

ćwiartki, to następujące punkty również należą do krzywej w pierwszym okresie: (PI – X, Y), (3 * PI/2 – X, Y), (2 * PI – X, –Y). Te symetryczne punkty można rysować kolejno dla tylu okresów, ile chcemy wyświetlić bez konieczności ponownego wyznaczania wartości z zależności (5.5).

Sinusoidy są użyteczne w zastosowaniach graficznych związanych z reprezentacją ruchu. Do zastosowań takich należą: symulacja głosu, muzyki, drgań sprężyny, odbić piłki, drgań wahadła. W przypadku sprężyny lub piłki musimy uwzględnić również tarcie – w każdym okresie zmniejsza się amplituda ruchu. To zmniejszenie się amplitudy możemy modelować mnożąc równanie (5.5) przez funkcję wykładniczą EXP (–K * X). Stała K określa szybkość zmniejszania się amplitudy. Dla K = 0,1 amplituda będzie się zmniejszała o około połowę w każdym okresie. Funkcje SIN i EXP są dostępne we wszystkich wersjach Basicu.

Krzywe wielomianowe

Ta klasa krzywych obejmuje w zasadzie nieskończoną listę zależności. Zależności te mają podobną strukturę podstawową i obejmują linie proste i parabole.

Równanie linii prostej ma postać

$$Y = C1 * X + C2 \tag{5.6}$$

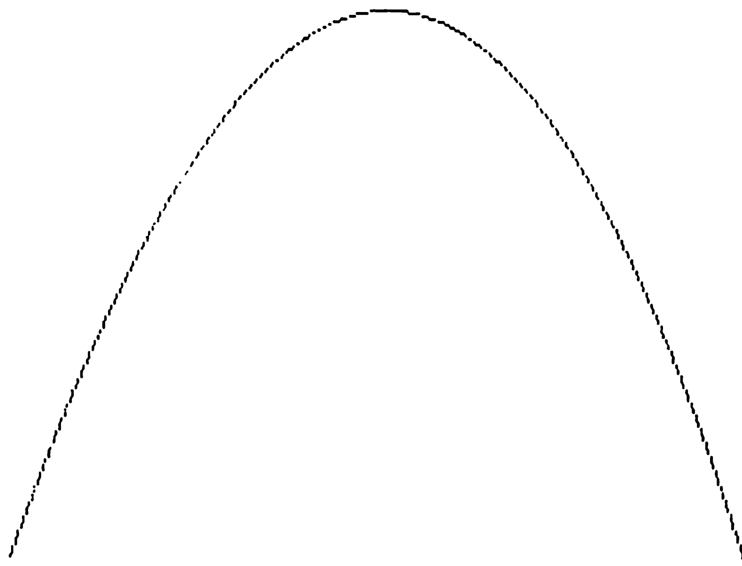
przy czym stałe C1 i C2 są odpowiednio nachyleniem i przesunięciem linii i są nazywane współczynnikami równania. Linia prosta jest klasyfikowana jako wielomian pierwszego stopnia. Dodanie do tego równania członów zawierających wyższe potęgi X prowadzi do wielomianów wyższego stopnia. Wielomian drugiego stopnia (parabola) ma postać

$$Y = C1 * X\hat{\ }2 + C2 * X + C3 \tag{5.7}$$

Parabole mogą być wykorzystywane do aproksymacji krzywych określonych przez stablicowane dane (w celu uzyskania informacji o tendencjach zmian lub interpolacji) lub do modelowania ruchów obiektów. Tor (trajektorię) piłki rzuconej pod pewnym kątem opisuje parabola. Piłka wznosi się do pewnej maksymalnej wysokości, a potem spada na ziemię. Wysokość jest mierzona za pomocą współrzędnej Y; odległość poziomą reprezentuje współrzędna X. Współrzędna X najwyższego punktu toru piłki

$$X = -C2/(2 * C1) \tag{5.8}$$

Na rysunku 5.13 podano wykres paraboli uzyskany za pomocą programu 5.5. Krzywa ta jest symetryczna względem wartości X określonej równaniem (5.8); dlatego program 5.5 oblicza punkty tylko dla połowy zakresu X.



Rys. 5.13. Parabola kreślona za pomocą programu 5.5

Program 5.5. Rysowanie paraboli

```
10  'PROGRAM 5. 5. Rysowanie paraboli
20      'Rysowana jest środkowa część paraboli (tzn. ta część gdzie
30      'krzywa zmienia nachylenie). Dla wierzchołka przyjmuje się
40      'Y = 0 albo Y = YM (zależnie od parametrów paraboli)
50      'i X = XM/2. Oblicza się wartości Y dla lewej części krzywej
60      '(kolejno zmniejsza się wartości X) i rysuje się odpowiednie
70      'punkty równocześnie z symetrycznymi punktami prawej części
80      'krzywej. W celu uzyskania ciągłości krzywej między sąsiednimi
90      'punktami rysuje się odcinki.
100     '**************************************************************
110 CLEARSCREEN
120 PRINT "Program rysuje parabolę o równaniu"
130 PRINT
140 PRINT "   Y = C1 * X ^ 2 + C2 * X + C3"
150 PRINT
160 PRINT "Jeżeli C1 jest mniejsze niż 0, krzywa rośnie do punktu naj-"
170 PRINT "wyższego,a następnie opada. Jeżeli C1 jest większe niż 0,"
180 PRINT "krzywa maleje do punktu najniższego,a następnie wznosi się."
190 PRINT
200 PRINT "Podaj wartości C1, C2, C3. (C1 nie może być zerem)"
210 INPUT C(1), C(2), C(3)
220 IF C(1) = 0 PRINT "Wprowadzono C1 = 0":GOTO 200
230 PRINT
240 PRINT "Podaj wartości maksymalna poziomą i pionową"
250 PRINT "dla przyjętej rozdzielczości"
260 INPUT XM, YM
270 XC = INT(XM / 2)        'XC jest w środku ekranu
280     'Wyznacz wartość X dla wierzchołka
290 X = - C(2) / (2 * C(1))
300     'Wyznacz wartość Y dla wierzchołka
310 YV = C(1) * X * X + C(2) * X + C(3)
320     'Czy ta krzywa ma maksimum czy minimum?
330     'Jeżeli maksimum Y1 = 0.Jeżeli minimum Y1 = YM.
340 IF C(1) < 0 THEN Y1 = 0
350 IF C(1) > 0 THEN Y1 = YM
360 XL1 = XC
370 XR1 = XC
380 XL2 = XC
390 XR2 = XC
400 CLEARSCREEN
410 GRAFHICS
420     'Oblicz punkty z lewej strony wierzchołka
430 X = X - 1
440     'Dla tej nowej wartości X znajdź Y rozwiązując równanie
450     'wielomianowe. W obliczeniach korzysta się z poniższej pętli.
460     'Można jej używać dla wielomianów dowolnego stopnia.
470 N = 3
480 Y = C(1)
490 FOR K = 2 TO N
500   Y = Y * X + C(K)
510 NEXT K
520     'Korekta wartości Y przed narysowaniem
530 IF C(1) < 0 THEN Y2 = YV - Y
540 IF C(1) > 0 THEN Y2 = YM - (Y - YV)
550 XL2 = XL2 -1
560 XR2 = XR2 +1
570     'Czy wszystkie wyznaczone punkty znajdują się na ekranie?
580 IF Y2 < 0 OR Y2 > YM OR XL2 < 0 OR XL2 > XM THEN 650
590 DRAWLINE XL1,Y1 TO XL2,Y2
600 DRAWLINE XR1,Y1 TO XR2,Y2
610 Y1 = Y2
620 XL1 = XL2
630 XR1 = XR2
640 GOTO 430
650 END
```

Program 5.5 rysuje dowolną parabolę określoną przez współczynniki C1, C2 i C3. Zależnie od wartości współczynnika C1 parabola będzie albo rosła do maksymalnej wartości Y, albo malała do minimalnej wartości Y dla środkowej wartości X (5.8). Wartość Y będzie maksymalna, jeżeli $C1 < 0$, i minimalna, jeżeli $C1 > 0$. Program rysuje parabolę w taki sposób, aby punkt środkowy krzywej znalazł się w środku ekranu na górze (dla $C1 < 0$) lub na dole (dla $C1 > 0$).

Krzywe wielomianowe wyższego stopnia można rysować za pomocą programu 5.5 uzupełnionego możliwością wprowadzania stopnia i współczynników równania

$$Y = C(1) * X\hat{}\ N + C(2) * X\hat{}\ (N-1) + ... + C(N-1) * X + \\ + C(N) \qquad (5.9)$$

Równania wielomianowe stopnia N mogą być użyteczne przy aproksymacji krzywej przez zbiór punktów. Dla danej tablicy punktów możemy narysować gładką krzywą przechodzącą przez te punkty, korzystając z wielomianu trzeciego lub czwartego stopnia. Rysowanie wielomianu może wymagać pewnych eksperymentów w celu ustalenia zakresu danych. Najpierw należy przejrzeć punkty krzywej po to, aby znaleźć zakres wartości X, potem określić skalę, aby uzyskać krzywą w odpowiednim polu ekranu.

Krzywe rozkładu normalnego

Równanie krzywej rozkładu normalnego (krzywej Gaussa nazywanej niekiedy krzywą dzwonową) ma postać

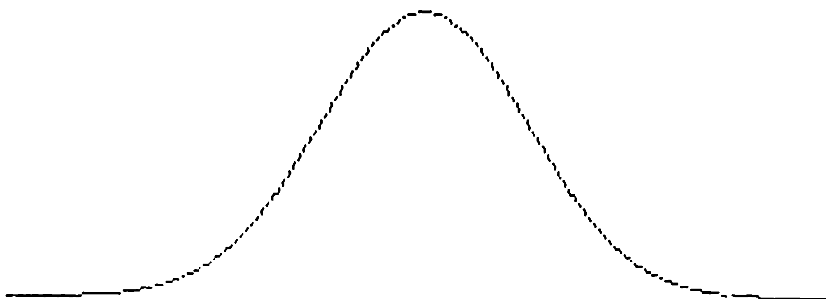
$$Y = EXP(-0{,}5 * (X - M)\hat{}\ 2/S\hat{}\ 2)/(S * SQR(2 * PI)) \qquad (5.10)$$

przy czym Y jest prawdopodobieństwem wystąpienia określonej wartości X w zbiorze danych. M jest średnią wszystkich wartości X, a S odchyleniem standardowym.

Równanie (5.10) opisuje przykładowy rozkład prawdopodobieństwa. Jest wiele innych krzywych opisujących rozkłady prawdopodobieństw, jednak największe znaczenie ma krzywa rozkładu normalnego, ponieważ wiele powszechnie spotykanych zjawisk w przybliżeniu daje się opisać za pomocą tego typu rozkładu. Prawdopodobieństwo, że pracownik w dużym przedsiębiorstwie otrzyma określoną pensję X, można oszacować jako Y w tym równaniu. Wartość $X = M$ jest średnią pensją wszystkich pracowników, a wartość S jest miarą rozrzutu pensji poniżej i powyżej średniej. Około 68% zatrudnionych otrzymuje pensje w zakresie jednego odchylenia standardowego od średniej ($M - S$ do $M + S$) i ok. 99% będzie miało pensje w zakresie 3 odchyleń standardowych od średniej ($M - 3 * S$ do $M + 3 * S$). Do innych zastosowań krzywej rozkładu normalnego należy wyświetlanie rozkładów prawdopodobieństw np.: czasu życia części elektrycznych lub mechanicznych, odchyłek wymiarów produkowanych wyrobów, wysokości kobiet w danym przedziale wieku, wielkości zysku z zainwestowanego kapitału lub zmiany temperatury w ciągu dnia w pewnym mieście.

Program 5.6. Rysowanie krzywej rozkładu normalnego

```
10  'PROGRAM 5.6. Rysowanie krzywej na podstawie zbioru próbek (do 500
20       'próbek)
30       'Krzywa jest umieszczona centrycznie względem osi X.Wartości Y
40       'leżą w przedziale 50 do 150.Obliczane i rysowane są punkty le-
50       'wej gałęzi krzywej. Prawa gałąź jest rysowana symetrycznie.
60       '**************************************************************
70  CLEARSCREEN
80  DIM D(500)
90  PI = 3.14159
100 PRINT "Podaj maksymalną wartość poziomą dla danej rozdzielczości"
110 INPUT XM
120 XC = XM / 2           'XC - środek osi X
130 PRINT "Podaj liczbę próbek"
140 INPUT N
150       'Znajdź wartość średnią
160 PRINT "Wprowadzaj po jednej próbce"
170 T = 0
180 FOR K = 1 TO N
190   INPUT D(K)
200   T = T + D(K)
210 NEXT K
220 M = T / N              'M - Wartość średnia danych
230       'Oblicz odchylenie standardowe
240 T = 0
250 FOR K = 1 TO N
260   T = T + (D(K) - M) ^ 2
270 NEXT K
280 V = T / N              'V - Wariancja
290 SD = V ^ .5            'SD - odchylenie standardowe
300       'Wyznacz punkt centralny krzywej
310       'Y ma wartość maksymalną,gdy X = M. Argumentem dla funkcji EXP
320       'jest 0 i EXP(0) = 1
330 X = M
340 YV = 1 / (SD * SQR(2 * PI))
350       'Przedział wartości punktów krzywej (0-YV) jest przeskalo-
360       'wywany na przedział 50-150 pikseli wzdłuż osi Y
370 YS = (150 - 50) / (YV - 0)
380       'i przeskalowywany na przedział 0-XM wzdłuż osi X
390 XS = (XM - 0) / (8 * SD)
400 XL1 = XC
410 XR1 = XC
420 Y1 = 50
430 CP = SD * SQR(2 * PI)      'Obliczanie stałej części równania
440       'Oblicz punkty lewej części krzywej. Rysuj obie części.
450 CLEARSCREEN
460 GRAPHICS
470 FOR X = M-1 TO M-4*SD STEP -1
480   Y = EXP(-.5 * (X - M) ^ 2 / V) / CP
490   'Koryguj Y tak,aby znalazł się w przedziale 50-150
500   Y2 = (YV - Y) * YS = 50
510   'Koryguj X tak,aby znalazł się w przedziale 0-XM
520   XL2 = XC - ((M - X) * XS)
530   XR2 = XC + ((M - X) * XS)
540   DRAWLINE XL1,Y1 TO XL2,Y2
550   DRAWLINE XR1,Y1 TO XR2,Y2
560   'Zapamiętaj te punkty
570   XL1 = XL2
580   XR1 = XR2
590   Y1 = Y2
600 NEXT X
610 END
```



Rys. 5.14. Krzywa rozkładu normalnego narysowana dla M = 51,8 i S = 29,2 za pomocą programu 5.6

Na rysunku 5.14 pokazano kształt krzywej rozkładu normalnego. Obraz ten uzyskano za pomocą programu 5.6, który dla zbioru danych wejściowych oblicza M i S oraz rysuje krzywą normalną. Wartości M i S są obliczane za pomocą standardowego równania. Krzywe rozkładu normalnego są symetryczne względem wartości średniej (X = M) i program 5.6 musi obliczyć punkty tylko dla połowy krzywej. Krzywa jest centrowana na ekranie; wartości Y są skalowane tak, aby zawierały się w zakresie 50 do 150 dla przedziału 4 standardowych odchyleń.

W tym punkcie dokonaliśmy przeglądu najczęściej używanych krzywych. Istnieje wiele innych krzywych, które mogą być użyteczne w zastosowaniach graficznych. Z krzywych Beziera i B-spline korzysta się przy wyświetlaniu powierzchni trójwymiarowych, na przykład w projektowaniu sylwetek samochodów czy samolotów. Funkcje Legendre'a i Bessela stosuje się w modelowaniu systemów fizycznych, takich jak struktury atomowe czy cząsteczkowe, rozkłady temperatur lub pola grawitacyjne. Z rozkładów prawdopodobieństw Poissona i hipergeometrycznego korzysta się w modelowaniu statystycznym, na przykład w symulacji kolejek klientów w banku przy różnych liczbach kasjerów lub przy analizie różnych wariantów wyboru zespołów projektowych z grupy pracowników.

5.3. Rysowanie obrazów z wykorzystaniem krzywych

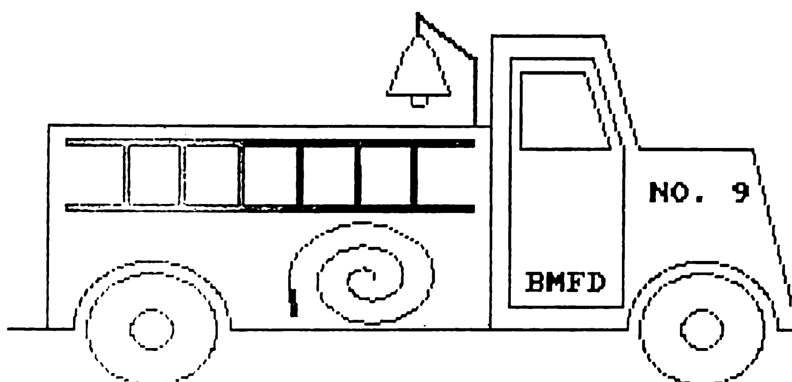
W tym punkcie podamy przykłady rysowania obrazów z wykorzystaniem krzywych. Za pomocą programu 5.7 otrzymujemy rys. 5.15 złożony z krótkich odcinków i opracowany na podstawie szkicu na papierze milimetrowym. Na rysunku 5.16 przedstawiono samochód straży pożarnej uzyskany za pomocą programu 5.8. Rysunek zawiera okręgi, spiralę i krzywą rozkładu normalnego. Program 5.9 jest przykładem „sztuki" graficznej uzyskanej za pomocą krzywych. Program ten demonstruje niektóre z wielu możliwości, jakie daje stosowanie funkcji trygonometrycznych. Uzyskane wzory pokazano na rys. 5.17.

Program 5.7. Dinozaur narysowany za pomocą krzywych aproksymowanych krótkimi odcinkami

```
10  'PROGRAM 5.7. Rysunek dinozaura
20       'Krzywe są aproksymowane krótkimi odcinkami.Punkty otrzymano z
30       'rysunku na papierze milimetrowym.Założone wymiary ekranu:
35       '640 x 200 pikseli.
40  CLEARSCREEN
50  GRAPHICS
60  R = 5/6
70       'Rysuj kolejne części obrazu
80  READ N            'Wczytaj liczbę punktów danej części
90  IF N = 0 THEN 420
100 READ X1, Y1
110 Y1 = Y1 * R       'Korekta wartości Y ze względu na rozdzielczość
120 FOR P =2 TO N
130   READ X2,Y2
140   Y2 = Y2 * R
150   DRAWLINE X1,Y1 TO X2,Y2
160   X1 = X2
170   Y1 = Y2
180 NEXT P
190 GOTO 80
200      'Dane dla głównej części ciała
210 DATA 38,223,60,180,71,173,65,174,63,179,64,180,62,185,62,187,60
220 DATA 192,60,194,57,200,57,202,55,183,55,182,59,172,58
230 DATA 171,62,165,57,170,50,220,31,233,31,245,40,260,60,290,67,315,72
240 DATA 345,80,360,87,380,105,400,127,425,145,452,155,502,164,450,164
250 DATA 434,162,415,158,395,153,362,138,323,132,283,125
260      'Dane dla jednej tylnej łapy
270 DATA 16,289,92,250,120,250,135,260,157,233,163,247,163,240,165
280 DATA 257,164,256,166,272,159,270,153,278,140,280,120,285,120
290 DATA 310,120,330,112
300      'Dane dla drugiej tylnej łapy
310 DATA 11,295,127,303,158,278,162,288,162,282,163
320 DATA 291,164,283,165,315,162,315,157,325,137,323,132
330      'Dane dla przedniej łapy
340 DATA 11,238,77,220,87,192,99,192,102,195,100
350 DATA 198,102,200,100,203,102,205,98,218,96,255,83
360      'Dane do wypełnienia
370 DATA 3,258,114,245,105,228,93
380 DATA 3,222,86,215,77,210,65
390      'Dane dla oka
400 DATA 4,220,42,210,45,215,40,220,42
410 DATA 0
420 END
```



Rys. 5.15. Obraz uzyskany za pomocą programu 5.7 na podstawie szkicu na papierze milimetrowym. Krzywe aproksymuje się odcinkami linii prostych



Rys. 5.16. Obraz uzyskany za pomocą programu 5.8, w którym wykorzystano równania krzywych

Program 5.8. Samochód straży pożarnej rysowany z wykorzystaniem równań linii krzywych

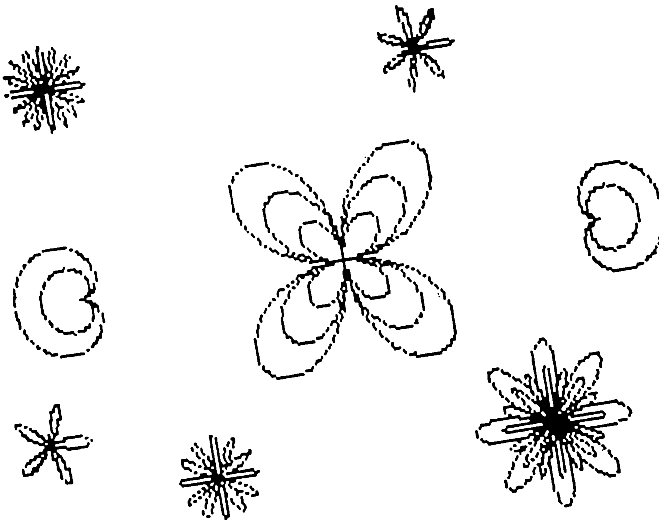
```
10  'PROGRAM 5.8. Samochód straży pożarnej
20       'W programie wykorzystuje się odcinki, okręgi, łuki, krzywą
25       'dzwonową i spiralę.
30       '****************** Rysuj kontur samochodu ******************
40 CLEARSCREEN
50 GRAPHICS
60 READ X1,Y1
70 FOR K = 1 TO 8
80   READ X2,Y2
90   DRAWLINE X1,Y1 TO X2,Y2
100  X1 = X2
110  Y1 = Y2
120 NEXT K
130      'Drzwi i okno
140 FOR M = 1 TO 2
150   READ N
160   READ X1,Y1
170   FOR K = 1 TO N
180     READ X2,Y2
190     DRAWLINNE X1,Y1 TO X2,Y2
200     X1 = X2
210     Y1 = Y2
220   NEXT K
230 NEXT M
240      '******** Wykończ kontur, drabinę, zawieszenie dzwonu *******
250 FOR L = 1 TO 13
260   'Wczytaj współrzędne, grubość, kierunek pogrubiania
270   READ X1, Y1, X2, Y2, T, D$
280   IF D$ = "X" THEN 350
290   'Pogrubienie w kierunku Y
300   FOR K = 0 TO T - 1
310     DRAWLINE X1,Y1+K TO X2,Y2+K
320   NEXT K
330   GOTO 390
340   'Pogrubienie w kierunku X
350   FOR K = 0 TO T - 1
360     DRAWLINE X1+K,Y1 TO X2+K,Y2
370   NEXT K
380   Y1 = Y2
390 NEXT L
```

Program 5.8 (cd.)

```
400   '******************** Wgłębienia na koła ***********************
410 FOR K = 1 TO 2
420   READ XC, YC, R
430   FOR A = 3.14159 TO 6.28318 STEP 1/R
440     XW = XC + R * COS(A)
450     YW = YC + R * SIN(A)
460     POINTPLOT XW,YW
470   NEXT A
480   COLOR 0,0
490   DRAWLINE XC-R+1,YC TO XC+R-1,YC        'Usuń część konturu
500   COLOR 1,0
510 NEXT K
520   '*************** Opony i kołpaki *******************************
530 FOR K = 1 TO 4
540   READ X,Y,R
550   CIRCLEPLOT X,Y,R
560 NEXT K
570   '********************** Wąż gaśniczy *************************
580 R = 1
590 A = 0.01
600 READ XC, YC
610 X1 = XC + R * COS(-A) * 1.4      'Zagęść spiralę w kierunku X
620 Y1 = YC + R * SIN(-A)
630 A = A + 0.1
640 R = R + 0.13
650 X2 = XC + R * COS(-A) * 1.4
660 Y2 = YC +R *SIN(-A)
670 DRAWLINE X1,Y1 TO X2,Y2
680 X1 = X2          'Zapamiętaj bieżące punkty
690 Y1 = Y2
700 IF R < 22 THEN 630
710    '****************** Napisy ************************************
720 POSITION 15,33
730 PRINT "NO. 9"
740 POSITION 19,27
750 PRINT "BMFD"
760    '****************** Dzwon *************************************
770 READ M ,SD
780 CP = SD * SQR(6.28318)
790 YV = 1 / CP
800 RY = (90-60) / YV
810 RX = (186 -159) / (3.4*SD)
820 XL1 = 166
830 XR1 = 166
840 Y1 = 60
850 FOR X = M-1 TO M-1.7*SD STEP -1
860   Y = EXP(-.5 * (X-M) ^ 2 / (SD * SD)) / CP
870   Y2 = (YV - Y) * RY + 60
880   XL2 = 166 - ((M - X) * RX)
890   XR2 = 166 + (M-X) * RX
900   DRAWLINE XL1,Y1 TO XL2,Y2
910   DRAWLINE XR1,Y1 TO XR2,Y2
920   XL1 = XL2
930   XR1 = XR2
940   Y1 = Y2
950 NEXT X
960 DRAWLINE XL1,Y1 TO XR1,Y1
970 CIRCLEPLOT 166,Y1+2,3          'Serce dzwonu
980    '*************************************************************
990 DATA 12,164,313,164,295,100,251,100,237,60,194,60,194,92,27,92,27
1000 DATA 164,5,201,68,201,156,245,156,245,100,233,68,201,68
1010 DATA 4,205,73,205,100,240,100,230,73,204,73
1020 DATA 194,60,194,164,1,X,34,100,187,100,3,Y,34,124,187,124,3,Y
1030 DATA 55,103,55,124,3,X,76,103,76,124,3,X,98,103,98,124,3,X
```

Program 5.8 (cd.)

```
1040 DATA 120,103,120,124,3,X,142,103,142,124,3,X,163,103,163,124,3,X
1050 DATA 187,92,187,68,2,X,187,68,165,52,2,Y,165,52,165,60,2,X
1060 DATA 125,153,126,159,3,X
1070 DATA 66,164,27,276,164,25,66,164,22,276,164,22,66,164,8,276,164,8
1080 DATA 147,147,
1090 DATA 1,7
1100 END
```



Rys. 5.17. Przykłady sztuki graficznej uzyskane za pomocą programu 5.9

Program 5.9. Wzory otrzymane za pomocą linii krzywych

```
10  'PROGRAM 5.9. Różne wzory otrzymane za pomocą krzywych
20  CLEARSCREEN
30  XM = 319
40  YM = 199
50  GRAPHICS
60        '******************** Koniczyna ***************************
70  XC = 160          'Srodek ekranu
80  YC = 100
90  FOR R = 20 TO 50 STEP 15
100   X1 = XC
110   Y1 = YC
120   FOR A = 0 TO 6.28318 STEP 1/R
130     R1 = R * SIN(2 * A)
140     X2 = XC + R1 * COS(A)
150     Y2 = YC + R1 * SIN(A)
160     DRAWLINE X1,Y1 TO X2,Y2
170     X1 = X2     'Zapamiętaj bieżące punkty
180     Y1 = Y2
190   NEXT A
200 NEXT R
210       '
220       '***************** Boczne figury (kardioidy) *************
230 XL = 60           'XL,YC - środek lewej figury
240 YC = 100
```

Program 5.9 (cd.)

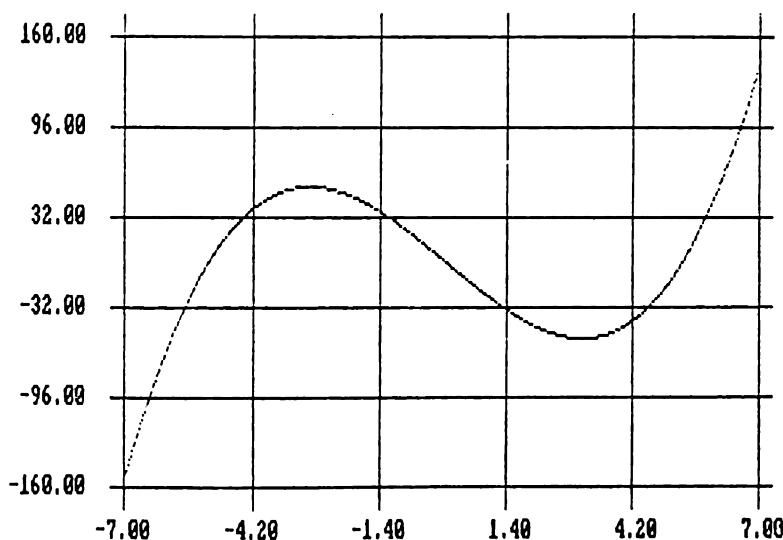
```
250 XR = 260        'XR,YC - środek prawej figury
260 FOR R = 15 TO 25 STEP 10
270   XL1 = XL
280   XR1 = XR
290   XL1 = YC
300   YL3 = YC
310   YR1 = YC
320   YR3 = YC
330   FOR A = 0 TO 3.14159 STEP 1/R
340     R1 = R * SIN(A / 2)
350     DX = R1 * COS(A)
360     DY = R1 * SIN(A)
370     XL2 = XL + DX
380     YL2 = YC + DY
390     YL4 = YC - DY
400     DRAWLINE XL1,YL1 TO XL2,YL2
410     DRAWLINE XL1,YL3 TO XL2,YL4
420     XR2 = XR - DX
430     YR2 = YC + DY
440     YR4 = YC - DY
450     DRAWLINE XR1,YR1 TO XR2,YR2
460     DRAWLINE XR1,YR3 TO XR2,YR4
470     XL1 = XL2
480     YL1 = YL2
490     YL3 = YL4
500     XR1 = XR2
510     YR1 = YR2
520     YR3 = YR4
530   NEXT A
540 NEXT R
550       '
560       '***************** Wzory kwiatowe ***************************
570 READ XC,YC,R,P        'Srodek,promień i liczba płatków
580 IF XC = 0 THEN 820
590 GOSUB 620
600 GOTO 570
610 GOTO 820
620 '************************ Rysuj kwiaty ****************************
630 X1 = XC + R
640 Y1 = YC
650 FOR A = 0 TO 6.28318 STEP 1/R
660   R1 = R * COS(P * A)
670   X2 = XC + R1 * COS(A)
680   Y2 = YC + R1 * SIN(A)
690   DRAWLINE X1,Y1 TO X2,Y2
700   X1 = X2
710   Y1 = Y2
720 NEXT A
730 RETURN
740 '*****************************************************************
750 DATA 200,30,15,7
760 DATA 100,170,15,6
770 DATA 60,25,15,8
780 DATA 40,150,14,5
790 DATA 230,169,20,8
800 DATA 230,169,30,4
810 DATA 0,0,0,0,0,0,0
820 END
```

5.4. Wykresy

Z krzywych można korzystać przy konstruowaniu różnego rodzaju wykresów. Ilustrują to programy przedstawione w tym punkcie.

Wykresy w układzie współrzędnych prostokątnych

Wykresy można konstruować dla punktów określonych w tablicach danych albo obliczanych z równań. W celu uzyskania wykresu możemy rysować punkty określone w tablicy danych i łączyć je odcinkami; można również korzystać z metod aproksymacji krzywych. Do rysowania krzywych przechodzących przez zadane punkty można korzystać z metod analitycznych (np. z metody najmniejszych kwadratów). Można również stosować metodę interakcyjnego rysowania krzywej, korzystając z instrukcji dostępnych z klawiatury lub z pióra świetlnego. Aby narysować wykres równania, obliczamy współrzędne z równania i rysujemy albo punkty leżące blisko siebie, albo odcinki łączące punkty.



Rys. 5.18. Wykres funkcji X 3 – 27 * X – wynik działania programu 5.10

Przykład rysowania wykresu równania zawiera program 5.10. Wymiary wykresu dla wybranej krzywej są podawane jako dane wejściowe do programu. Wynik działania programu dla wielomianu trzeciego stopnia pokazano na rys. 5.18.

Programu 5.10. Rysowanie wykresu dowolnego równania wejściowego

```
10  PROGRAM 5.10. Rysowanie wykresu dowolnego rownania
20       'Użytkownik może w wierszu 450 podać swoje równanie. Można po-
30       'dać największe i najmniejsze wartości X i Y,dla których należy
40       'rysować wykres. Program rysuje siatkę w prostokącie o bokach:
50       'w kierunku X - piksele 74-574, w kierunku Y piksele 12-188.
60       '**************************************************************
70  CLEARSCREEN
80  PRINT "Czy potrzebny komentarz? (T - tak,N - nie)"
90  INPUT I$
100 IF I$ ="N" THEN 220
110 PRINT "Program wyświetla wykres dowolnego równania na tle siatki"
120 PRINT "prostokątnej. Równanie należy wprowadzić w wierszu 450."
130 PRINT "Zmienną niezależną w równaniu musi być X, zmienną zależną Y"
140 PRINT "(np. Y = 6 * X + 20).Po wprowadzeniu równania i napisaniu"
150 PRINT "RUN pojawi się żądanie określenia najmniejszej i największej"
160 PRINT "wartości współrzędnej X określających zakres wykresu, oraz"
170 PRINT "najmniejszej i największej wartości Y określających skalę"
180 PRINT "siatki."
190 PRINT
200 PRINT
210 PRINT
220 PRINT "Czy w wierszu 450 jest właściwe równanie? (T - tak,N - nie)
230 INPUT E$
240 IF E$ = "T" THEN 300
250 IF I$ = "N" THEN 780
260 PRINT
270 PRINT "Napisz numer wiersza 450 oraz równanie"
280 PRINT "Naciśnij RETURN oraz napisz RUN"
290 GOTO 780
300      'Rysuj wykres równania na tle siatki
310 PRINT
320 PRINT "Wprowadź wartości X najmniejszą i największą,dla których"
330 PRINT "należy rysować wykres."
340 INPUT XL, XR
350 PRINT
360 PRINT "Wprowadź wartości Y najmniejszą i największą ,które powinny"
370 PRINT "pojawić się na siatce."
380 INPUT YB, YT
390 GOSUB 520        'Rysuj siatkę
400      'Rysuj wykres równania
410 XA = (564 - 84) / XD
420 YA = (180 - 20) / YD
430 FOR XG = 84 TO 564
440   X = XL + (XG - 84) / 484 * XD
450   Y = X ^ 3 - 27 * X
460   YG = 180 - (Y - YB) * YA
470   IF YG < 20 OR YG > 180 THEN 490
480   POINTPLOT XG,YG
490 NEXT XG
500 POSITION 1,1
510 GOTO 780
520      'Rysowanie siatki z opisem
530 CLEARSCREEN
540 GRAPHICS
550 FOR Y = 20 TO 180 STEP 32
560   DRAWLINE 74,Y TO 574,Y
570 NEXT Y
580 FOR X = 84 TO 564 STEP 96
590   DRAWLINE X,12 TO X,188
600 NEXT X
610 XD = XR - XL
620 YD = YT - YB
630 R = 23
```

Program 5.10 (cd.)

```
640 FOR K = 0 TO 5
650   POSITION R,1
660   S = YB + YD * K / 5
670   PRINT USING "####.##";S
680   R = R - 4
690 NEXT K
700 C = 7
710 FOR K = 0 TO 5
720   POSITION 25,C
730   S = XL + XD * K / 5
740   PRINT USING "####.##;S;
750   C = C +12
760 NEXT K
770 RETURN
760 END
```

Wykresy kołowe

Podstawowym składnikiem programu tworzącego wykresy kołowe jest algorytm rysowania okręgu. Zastosowanie takie ilustruje program 5.11. W danych

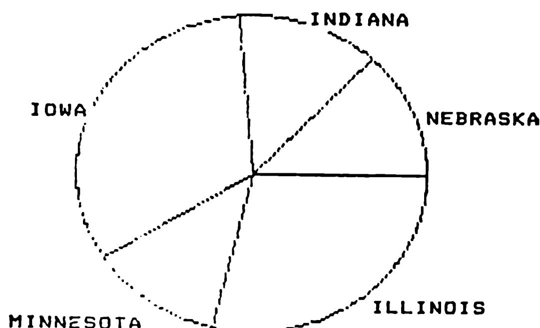
Program 5.11. Wykres kolowy

```
10  'PROGRAM 5.11. Opisany wykres kołowy
20  CLEARSCREEN
30  DIM N$(8), V(8)
40  PRINT "Podaj wartości maksymalne poziomą i pionową"
50  PRINT "dla przyjętej rozdzielczości"
60  INPUT XM, YM
70  PRINT "Wprowadź liczbę znaków,które można umieścić w wierszu"
80  PRINT "dla przyjętej rozdzielczości"
90  PRINT
100 INPUT C1
110 PC = XM / C1      'PC jest liczbą pikseli na znak w poziomie
120 PRINT "Podaj liczbę wierszy dostępnych"
130 PRINT "dla przyjętej rozdzielczości"
140 INPUT R1
150 PR = YM / R1      'PR jest liczbą pikseli na znak w pionie
160 PRINT
170 PRINT "Podaj współrzędne środka wykresu kołowego"
180 INPUT XC, YC
190 PRINT "Podaj promień"
200 INPUT R
210 IF XC + R > XM OR XC - R < 0 OR YC + R > YM OR YC - R < 0 THEN 830
220 PRINT
230 PRINT "Podaj liczbę części (do 8)"
240 INPUT N
250 PRINT "Podaj tytuł wykresu"
260 INPUT T$
270 PRINT "Podaj dla każdej części jej wielkość i nazwę"
280 T = 0
290        'Wprowadź dane. Znajdź sumę (T) wszystkich wartości
300 FOR K = 1 TO N
310   INPUT N$(K), V(K)
320   T = T + V(K)
330 NEXT K
340 CLEARSCREEN
350 GRAPHICS
360 POSITION 1,INT(C1 / 2 - 0.5 * LEN(T$))        'Umieść tytuł wykresu
370 PRINT T$                                      'w środku ekranu
380 CIRCLEPLOT XC,YC,R
390 YA = 5 / 6       'YA - współczynnik korekcji rozdzielczości
```

Program 5.11 (cd.)

```
400 B = 0            'B - kąt dla poprzedniej linii
410 S = 0
420 RE = 360 * 3.14159 / 180        'RE - odpowiednik kąta 360 stopni
430 FOR K = 1 TO N                  'w radianach
440      'Rysowana linia jest określana na podstawie wartości danej
450      'części i poprzednich części
460   S = S + V(K)
470      'Kąt określa się z proporcji między częścią koła
480      'a całym kołem (S / T)
490   A = RE * S / T
500   XP = XC + R * COS(A)
510   YP = YC + R * SIN(A) * YA
520   DRAWLINE XC, YC TO XP,YP
530      'Opis segmentu wykresu
540      'Wyznacz punkt na zewnątrz okręgu w odległości 4 pikseli
550      'od środka łuku ograniczającego dany segment
560   AC = B + (A - B) / 2              'AC jest kątem między pozio-
570   XL = XC + (R + 4) * COS(AC)       'mem a linią środkową okręgu
580   YL = YC + (R + 4) * SIN(AC) * YA
590      '(XL,YL) punkt bazowy dla opisu
600      'Opis leżący z prawej strony okręgu zaczyna się w punkcie ba-
610      'zowym; opis leżący z lewej strony okręgu kończy się w punkcie
620      'bazowym; punkt bazowy określa środek opisu nad lub pod okrę-
625      'giem
630      'Punkt bazowy jest punktem początkowym opisu
640   IF XL > XC + 10 THEN 740
650      'Punkt bazowy jest punktem końcowym opisu
660   IF XL < XC - 10 THEN 710
670      'W przeciwnym wypadku punkt bazowy określa środek opisu. Sko-
680      'ryguj XL o liczbę równą połowie liczby pikseli dla opisu
690   XL = XL - LEN(N$(K)) / 2 * PC
700   GOTO 740
710      'Punkt bazowy wyznacza koniec opisu. Cofnij o liczbę
720      'pikseli potrzebną do wykonania opisu
730   XL = XL - LEN(N$(K)) * PC
740      'Zamień pozycję piksela (XL,YL) na najbliższą
750      'pozycję znaku
760   RO = INT(YL / PR + .5) + 1
770   CO = INT(XL / PC + .5) + 1
780   POSITION RP,CO
790   PRINT N$(K)
800   B = A
810 NEXT K
820 GOTO 840
830 PRINT "Współrzędna poza zakresem"
840 END
```

wejściowych do programu określa się nazwę i względne wymiary (wartość) każde części wykresu. Miejsca wpisania nazw poszczególnych części określa się podając kąt dla linii środkowej odpowiedniego wycinka. Jeżeli wycinek jest z prawej strony wykresu, początek opisu zaczyna się w punkcie leżącym na linii środkowej w odległości 4 jednostek od obwodu. Jeżeli wycinek jest z lewej strony, opis kończy się na linii środkowej. Początkowe pozycje opisów znajdujemy zamieniając współrzędne pikseli na pozycje znaków. Uzyskujemy to, dzieląc współrzędną X przez liczbę pikseli w znaku w kierunku poziomym oraz dzieląc współrzędną Y przez liczbę pikseli w znaku w kierunku pionowym. Wykres kołowy pokazany na rys. 5.19 otrzymano w wyniku wykonania programu 5.11.



Rys. 5.19. Wykres kołowy – wynik działania programu 5.11

Ze względu na przejrzystość wykres kołowy nie powinien mieć więcej niż pięć lub sześć części. Zasady wyboru koloru i cieniowania wykresu kołowego są podobne jak dla wykresów słupkowych. Wzory cieniowania powinny być proste i powinno być zachowane stopniowanie jasności wokół wykresu. Opisy powinny być na obszarze, którego dotyczą, lub blisko niego.

Rodzaj użytego do prezentacji informacji wykresu może mieć duży wpływ na jakość prezentacji. Wykresy kołowe są najlepsze do przedstawiania informacji procentowych. Do tego celu można również stosować wykresy słupkowe dzieląc poszczególne słupki na sekcje odpowiadające procentom. Wykresy liniowe i słupkowe są dobre do ilustrowania informacji o takich danych liczbowych, jak np. wielkość sprzedaży. Wykres wielkości sprzedaży można zrobić w funkcji czasu albo w odniesieniu do jakiegoś innego parametru, np. regionu. O ile regiony sprzedaży mogą być związane albo z osią pionową, albo z osią poziomą, o tyle czas zazwyczaj najlepiej jest reprezentować na osi poziomej. Bardziej złożone zależności między danymi można przedstawiać za pomocą kilku krzywych na jednym wykresie, nakładając słupki na siebie w wykresie słupkowym lub korzystając z wykresu trójwymiarowego.

ZADANIA

5.1. Zmodyfikuj program 5.1 tak, żeby daną wejściową do rysowania okręgu była kątowa wartość kroku, a nie liczba punktów. Wtedy, dla kroku 120° program będzie wyświetlał trójkąt, a dla kroku 90° kwadrat.

5.2. Zaproponuj program rysowania łuku o zadanym promieniu i współrzędnych środka. W danych wejściowych należy określać współrzędne końców łuku lub kąty początkowy i końcowy.

5.3. Zaproponuj program wyświetlania półksiężyca za pomocą dwóch łuków o różnych promieniach.

5.4. Napisz program generacji koła, którego wnętrze jest kolorowane. Wnętrze koła wypełnij dowolnym kolorem metodą rysowania średnic o zadanym kolorze. Korzystając z równań (5.1) można wymazać punkty końcowe średnic pamiętając, że odpowiadające im kąty różnią się o 180°. Można uzyskać efekty specjalne zmieniając kolor kolejnych średnic.

5.5. Wnętrze koła można wypełnić punktami rozmieszczonymi w różnych odległościach od siebie, położonymi wzdłuż promieni od środka koła w kierunku obwodu. Napisz program rysowania koła tym sposobem. Wykorzystaj równanie (5.1) i zmieniaj promień od zera do R.

5.6. Napisz program wypełniania wnętrza koła za pomocą linii poziomych. Dla koła o środku w punkcie (XC, YC) współrzędne X punktów końcowych linii poziomych w kole są określone jako:

XC—SQR (R^2—(Y—YC)^2

XC+SQR (R^2—(Y—YC)^2

Współrzędne Y zmieniają się od YC — R do YC + R. Zmieniając odstęp między liniami i rysując punkty zamiast linii ciągłych uzyskuje się różne wzory cieniowania. (Malowanie za pomocą linii pionowych uzyskuje się dzięki zamianie wartości X i Y oraz XC i YC w obliczeniach).

5.7. Napisz program wyświetlania elipsy malowanej jedną z metod podanych w programach 5.4, 5.5 lub 5.6.

5.8. Zmodyfikuj program 5.4 tak, by rysował wykres sinusoidy z opisanymi osiami. Zmniejsz liczbę obliczeń korzystając z własności symetrii funkcji sinus. W programie wykorzystaj równanie (5.5). Dane wejściowe to: amplituda H, częstotliwość W, przesunięcie D. Narysuj trzy okresy krzywej w przedziale od X = −D/W do X = (6* PI − D)/W.

5.9. Program omówiony w zadaniu 5.8 zmodyfikuj tak, aby wyświetlał tłumioną funkcję sinus mnożąc jej wartość przez EXP(−K * X) dla każdej wartości X. Narysuj wynikową krzywą w przedziale od X = 0 do X = (10 * PI − D)/W dla pewnej dodatniej wartości wejściowej K.

5.10. Napisz program wyświetlania zbioru zadanych punktów jako małych kółek, przy czym współrzędne środka każdego kółka są określone przez poszczególne punkty. Dołącz opisane osie współrzędnych.

5.11. Napisz program interakcyjnego rysowania krzywej przechodzącej przez zadany zbiór punktów. Program powinien umożliwiać szkicowanie krzywej za pomocą klawiatury lub innego urządzenia do interakcyjnej współpracy.

5.12. Napisz program wyświetlania zbioru zadanych punktów oraz paraboli najlepiej aproksymującej krzywą przechodzącą przez te punkty. Dla zbioru N punktów wejściowych, (X(1), Y(1)), (X(2), Y(2)), ..., (X(N), Y(N)) współczynniki paraboli (C1, C2, C3 w równaniu (5.7)) można określić metodą najmniejszych kwadratów. W celu znalezienia C1, C2, C3 trzeba rozwiązać następujący układ równań:

∑ Y(I) = C1 * ∑ X(I)^2+C2 * ∑ X(I)+C3 * N

∑ X(I) * Y(I) = C1 * ∑ X(I)^3+C2 * ∑ X(I)^2+C3 * ∑ X(I)

∑ (X(I)^2) * Y(I) = C1 * ∑ X(I)^4+C2 * ∑ X(I)^3+C3 * ∑ X(I)^2

w których symbol ∑ (sigma) oznacza sumę dla wszystkich wartości I od 1 do N.

5.13. Narysuj obraz lub scenę na papierze milimetrowym i napisz program wyświetlania tego obrazu, korzystając z odpowiednich funkcji opisujących krzywe aproksymujące kontury figur. Kontur wypełnij kolorem lub odpowiednim wzorem cieniującym.

Część III
ŚREDNIO ZAAWANSOWANE TECHNIKI GRAFICZNE
(Co więcej możemy zrobić?)

Dotychczas skupiliśmy uwagę na podstawowych metodach grafiki komputerowej umożliwiających konstruowanie prostych obrazów. Zobaczyliśmy, w jaki sposób można tworzyć wykresy i obrazy różnego rodzaju. Teraz rozważymy sposoby manipulowania tymi obrazami. Zorientujemy się, w jaki sposób można modyfikować części obrazu i jak realizuje się animację.

6. Transformacje obrazów

Metody transformacji omawiane w tym rozdziale stanowią podstawowe narzędzia manipulowania obrazami. Zrozumienie zasad modyfikacji lub transformacji obrazu znacznie zwiększy nasze umiejętności tworzenia grafiki komputerowej. Do specyficznych zastosowań takich metod należą: przesuwanie obrazów i wykresów z jednego miejsca ekranu w inne, tworzenie złożonego obrazu z prostszych części, tworzenie dodatkowych elementów już opracowanego obrazu, powiększanie wymiarów obrazu lub wykresu w celu uzyskania lepszej jakości obrazu lub obejrzenia większej liczby szczegółów, zmniejszanie wymiarów obrazu w celu dodania pomocniczych informacji takich, jak opisy czy inne obrazy, animacja obrazów.

Takie przekształcanie prostych obrazów uzyskuje się za pomocą trzech podstawowych transformacji: 1) przesuwania wyświetlanego obiektu z jednego miejsca ekranu w inne (translacja), 2) zwiększania lub zmniejszania wymiarów obrazu lub wykresu (skalowanie), 3) zmiany kierunku orientacji reprezentacji graficznej (rotacja).

6.1. Zmiany położenia (przesunięcia)

Zmiana położenia na ekranie umożliwia tworzenie obrazu lub wykresu w dowolnej części ekranu, a następnie przesuwanie otrzymanego wyniku w inne miejsce na ekranie. Celem naszym może być jedynie przesunięcie obrazu. Możemy również budować obraz częściami, korzystając ze zbioru elementów składowych.

Przesuwanie punktu

Zmiana położenia wyświetlonego punktu wymaga jedynie zmiany współrzędnych punktu na ekranie na nowe współrzędne. W celu przesunięcia punktu w poziomie w prawo o 15 pikseli dodajemy 15 do bieżącej współrzędnej X. Jeżeli

chcemy również przesunąć punkt do góry ekranu o 40 pikseli, musimy odjąć 40 od bieżącej współrzędnej Y. Ogólna reguła przesuwania punktu na ekranie z miejsca (X, Y) do nowego miejsca (XT, YT) w zależności od *wielkości przesunięcia* H i V ma postać:

$$\begin{aligned} XT &= X + H \\ YT &= Y + V \end{aligned} \tag{6.1}$$

Dodatnia wartość H oznacza przesunięcie punktu poziomo w prawo, a dodatnia wartość V – przesunięcie punktu w dół ekranu. Ujemne wartości H lub V oznaczają przesunięcie punktu odpowiednio w lewo lub do góry.

Przesuwając punkt należy korzystać tylko z takich wartości H i V, które nie spowodują wyjścia punktu poza granice ekranu. Należy również unikać bardzo małych wartości H i V. Jeżeli obie wartości H i V są mniejsze niż 0,5, to początkowy punkt będzie po prostu ponownie narysowany. Wielokrotne rysowanie obrazu może w istotny sposób zmniejszyć szybkość programu.

Przesuwanie obrazu

Przesuwanie obrazu z jednego miejsca ekranu w inne oznacza, że przesuwamy wszystkie punkty obrazu, a następnie rysujemy linie łączące przesunięte punkty. Aby uniknąć zmiany kształtu obiektu, musimy przesuwać wszystkie punkty o taką samą odległość. To znaczy, dokonuje się obliczeń (6.1) z taką samą wartością H dla wszystkich współrzędnych poziomych i z taką samą wartością V dla wszystkich współrzędnych pionowych obrazu. Przyjmując różne wartości V lub H dla różnych punktów obrazu, zniekształcimy oryginalny obraz. Zazwyczaj chcemy przesuwać obiekty bez zniekształceń; niemniej dopuszczenie takich zakłóceń umożliwia eksperymentowanie z doborem kształtów w procesie projektowania, ma również zastosowanie w grach komputerowych.

Współrzędne początku i końca każdego odcinka w obrazie można zapamiętać w tablicy. Punkty prostego obrazu zapamiętuje się w tablicy jednowymiarowej w takiej kolejności, w jakiej chcemy je łączyć. W bardziej złożonym obrazie punkt może należeć do kilku odcinków, mogą również wystąpić części obrazu, które nie są połączone z innymi częściami, tak jak na rys. 6.1. Wtedy wartości współrzędnych zapamiętuje się w tablicy dwuwymiarowej. Jeden indeks tablicy identyfikuje część obrazu, a drugi określa, o który punkt tej części chodzi. I tak, (X (2, 1), Y (2, 1)) może być użyte do zapamiętania współrzędnych pierwszego punktu 2 części obrazu (np. pies na rys. 6.1). W programie 6.1 wykorzystano tablice dwuwymiarowe do pamiętania współrzędnych definiujących detale z rys. 6.1. Program umożliwia przesuwanie elementów obrazu do dowolnego miejsca na ekranie dowolną liczbę razy.

Program 6.1 zapamiętuje przesunięte współrzędne w oryginalnych tablicach X i Y. Jeżeli chcemy pamiętać początkową pozycję, to przesunięte współ-

rzędne zapisuje się w innych tablicach, np. XT i YT. Pamiętanie początkowej pozycji jest celowe, jeżeli chcemy tylko sprawdzić efekt wizualny przy nowym położeniu elementów obrazu lub też gdy chcemy wyświetlić obiekt w dwóch miejscach. Jeżeli nie ma potrzeby przechowywania informacji o początkowej pozycji obrazu, to oszczędzamy przestrzeń pamięci zapisując nowe współrzędne w pierwotnych tablicach X i Y.

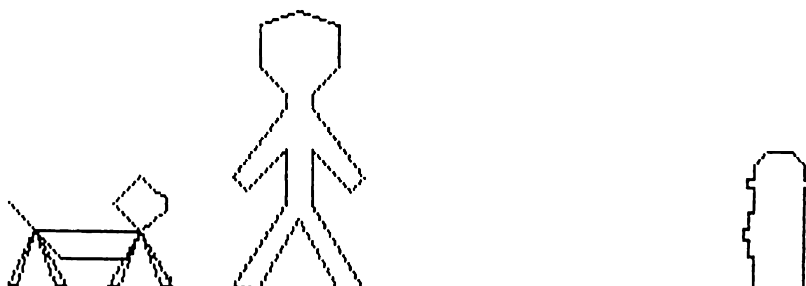
W przypadku obiektów symetrycznych lub gdy brzegi obrazu wyznaczamy z równań, do relokacji (przesunięcia) obrazu nie jest konieczne dodawanie wartości przesunięcia do każdego punktu. Na przykład, aby przesunąć okrąg lub elipsę, wystarczy przesunąć środek figury i obliczyć nowe punkty wzdłuż brzegu korzystając z programów omawianych w rozdz. 5. W celu przesunięcia prostokąta wystarczy przesunąć jeden wierzchołek i narysować prostokąt, korzystając z jego wymiarów: szerokości i wysokości.

Program 6.1. Przesuwanie obrazów (chłopiec, pies i hydrant)

```
10  'PROGRAM 6.1. Przesuwanie części obrazu
20      'Program rysuje obraz i umożliwia przesuwanie jego części w
30      'inne miejsca. Części obrazu są pamiętane w tablicach X i Y.
40      'Przesunięte punkty zastępują w tablicach X i Y punkty
50      'oryginalne.
60  CLEARSCREEN
70  DIM X(5,50), Y(5,50), N(5)
80  XM = 319        'XM,YM - maksymalne wartości pikseli w tym systemie
90  YM = 199
100     '*********** Czytaj i rysuj części obrazu ******************
110 PN = 1          'PN - numer części
120 E = 0
130 READ XD, YD
140 IF XD >= 0 THEN 200
150     'Zapamiętaj liczbę elementów w PN w N(PN)
160 N(PN) = E
170 IF XD = -100 THEN 240
180 PN = PN + 1
190 GOTO 120
200 E = E + 1
210 X(PN,E) = XD
220 Y(PN,E) = YD
230 GOTO 130
240 GRAPHICS
250 GOSUB 510       'Rysuj obraz
260     '
270     '***************** Instrukcje drukowania *********************
280 POSITION 22,1
290 PRINT "          1-Chłopiec   2-Pies   3-Hydrant"
300 PRINT "Jaką część rysunku przesunąć (napisz 0 aby zakończyć)?";
310 INPUT P
320 POSITION 22,1
330 PRINT "                                                    "
340 PRINT "                                                    ";
350 IF P = 0 THEN 730
360 POSITION 23,1
370 PRINT "Podaj wartości przesunięcia H i V";
380 INPUT H, V
390     '
400     '***************** Przelicz punkty *************************
410 FOR J = 1 TO N(P)
420   X(P,J) = X(P,J) + H
430   Y(P,J) = Y(P,J) + V
```

Program 6.1 (cd.)

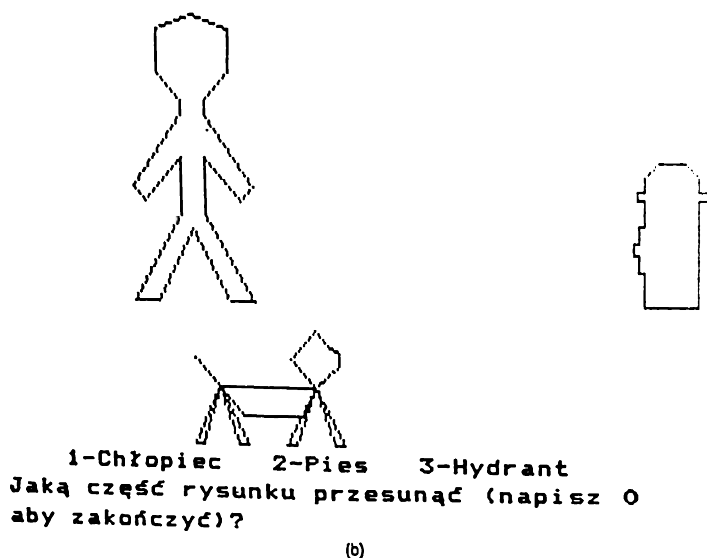
```
440   IF X(P,J)>=0 AND X(P,J)<=XM AND Y(P,J)>=0 AND Y(P,J)<=YM THEN 470
450   PRINT "Współrzędna poza zakresem"
460   GOTO 730
470 NEXT J
480 GOSUB 510
490 GOTO 280
500 '
510 '********************** Program rysowania ***********************
520 CLEARSCREEN
530 FOR K = 1 TO PN
540   FOR J = 1 TO N(K)-1
550     DRAWLINE X(K,J), Y(K,J) TO X(K,J+1), Y(K,J+1)
560   NEXT J
570 NEXT K
580 RETURN
590 '****************************************************************
600 DATA 85,70,90,75,105,60,105,80,85,110,95,110,110,85
610 DATA 125,110,135,110,115,80,115,60,130,75,135,70,115,45
620 DATA 115,40,125,30,125,15,110,10,95,15,95,30,105,40,105,45,85,70
630 DATA -1,-1
640 DATA 50,90,62,110,58,110,50,90,42,110,38,110,50,90
650 DATA 45,100,20,100,10,90,22,110,18,110,10,90,3,110
660 DATA 0,110,10,90,0,80,10,90,50,90,40,80,50,70,55,75,60,78,60,82,50,90
670 DATA -1,-1
680 DATA 290,110,290,98,288,98,288,92,286,92,286,88
690 DATA 286,88,288,88,288,82,290,82,290,73,287,73,287,70,290,70,290,65
700 DATA 295,60,305,60,310,65,310,70,313,70,313,73,310,73,310,110,290,110
710 DATA -100,-100
720 '****************************************************************
730 END
```



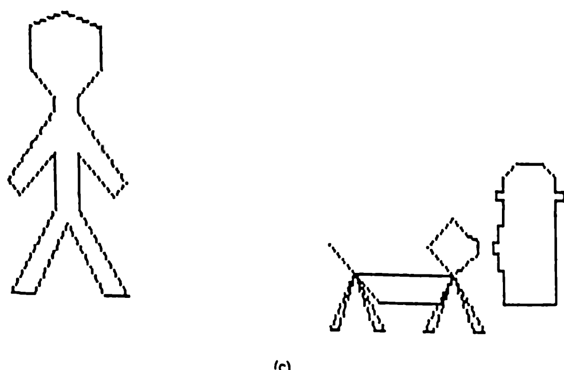
1-Chłopiec 2-Pies 3-Hydrant
Jaką część rysunku przesunąć (napisz 0
aby zakończyć)?

(a)

Rys. 6.1a. (podpis na stronie 112)



(b)



(c)

Rys. 6.1. Przesuwanie elementów obrazu z oryginalnej pozycji (a) do pozycji (b), a następnie do pozycji (c) za pomocą programu 6.1

Przesuwanie wykresu

Przedstawione metody i rozważania związane z przesuwaniem obrazów stosuje się również do wykresów – wszystkie punkty wykresu otrzymują nowe wartości (w granicach ekranu) i wszystkie linie są na nowo rysowane w nowym miejscu. Opis może być utworzony ponownie w odpowiednich miejscach nowego wykresu albo przeniesiony na nowe pozycje razem z innymi współrzędnymi punktów obrazu.

Aby przenieść znaki opisu, wartości przesunięcia H i V należy wyrazić w postaci różnic pozycji znaków. Wartość przesunięcia opisu w poziomie otrzymujemy dzieląc H przez liczbę poziomych punktów w siatce znaku. Przesunięcie w pionie otrzymujemy dzieląc V przez liczbę pionowych punktów w siatce znaku. Dla siatki znaku 8 × 12 pikseli numer wiersza opisu powinien być zmieniony o wartość V/12, a pozycja w poziomie powinna być zmieniona o H/8. Przewidując przesunięcie opisów powinniśmy tak wybierać wartości H i V, aby były wielokrotnościami wymiarów siatki znaku. Umożliwi to zachowanie po przesunięciu położenia opisów względem wykresu.

Sposób przesuwania wykresu słupkowego podano w programie 6.2. Przy przesuwaniu opisów korzystano z siatki znaku 8 × 12 pikseli. W celu określenia długości każdego opisu przy przesuwaniu i sprawdzenia, czy nie przekracza się granic ekranu, posłużono się funkcją LEN.

Program 6.2. Przesuwanie wykresu

```
10  'PROGRAM 6.2. Przesuwanie opisanego wykresu słupkowego poziomego
20      'Założone wymiary ekranu: 208 x 160 pikseli, 15 wierszy po 35
30      'znaków w wierszu (każdy znak zajmuje powierzchnię 8 x 12
40      'pikseli). Zmienna liczba (do 8) opisów, wymiary i nazwa wy-
50      'kresu są wczytywane za pomocą instrukcji DATA. Opisy są pamię-
60      'tane w tablicy L$. Pozycja każdego opisu jest pamiętana w ta-
70      'blicach R i C. Długość opisu nie może przekraczać 8 znaków.
80      'Wymiary wykresu są określone przez przedział 64-160 pikseli.
90      'Punkty końcowe słupków dla każdej części wykresu są pamiętane
100     'w tablicach X i Y. Po utworzeniu wykresu można go przesunąć do
110     'innego miejsca na ekranie. Przesunięte punkty pamięta się w
120     'nowych tablicach.
130 '
140 '****************************************************************
150 CLEARSCREEN
160 DIM X(8,2), Y(8,2), L$(9), R(9), C(9)
170 DIM XT(8,2), YT(8,2), LT$(9), RT(9), CT(9)
180     '
190     '************* Odczytaj dane dla wykresu i rysuj **************
200 READ MI, MA          'Odczytaj wartości najmniejszą i największą
210 RM = (160 - 64) / (MA - MI)    'w celu określenia zakresu danych
220     'Czytaj dane dla wykresu
230 READ N               'Odczytaj liczbę części
240 FOR K = 1 TO N
250   READ L$(K),M
260   R(K) = K + 2         'Zostaw pierwsze dwa wiersze na tytuł i odstęp
270   C(K) = 1
280   X(K,1) = 64
290   X(K,2) = INT((M - MI) * RM + 64 + 0.5)
300   Y(K,1) = (K + 1) * 8 + 1          'Góra słupka
310   Y(K,2) = Y(K,1) + 4               'Dół słupka
320 NEXT K
330 READ L$(N+1)         'Odczytaj tytuł
340 R(N+1) = 1
350 C(N+1) = 10 - LEN(L$(N+1)) / 2     'Środek tytułu w 10 kolumnie
360     'Kopiuj oryginalne wartości do tablic używanych do przechowy-
370     'wania przesuniętych punktów. Z tablic tych korzysta się przy
375     'rysowaniu wykresu
```

Program 6.2 (cd.)

```
380 FOR K = 1 TO N
390   RT(K) = R(K)
400   CT(K) = C(K)
410   FOR J = 1 TO 2
420     XT(K,J) = X(K,J)
430     YT(K,J) = Y(K,J)
440   NEXT J
450 NEXT K
460 RT(N+1) = R(N+1)
470 CT(N+1) = C(N+1)
480 GRAPHICS
490 GOSUB 830             'Rysuj wykres
500      '
510      '****************** Instrukcje drukowania ********************
520 POSITION 20,1
530 PRINT "Podaj wartości przesunięcia H i V";
540 PRINT "od pozycji początkowej. Wprowadź 0,0 aby zobaczyć pozycję";
550 PRINT "początkową, -999, -999 aby zakończyć program";
560 INPUT H,V
570      'Instrukcje usuwania
580 POSITION 20,1
590 PRINT "                                              ";
600 PRINT "                                              ";
610 PRINT "                                              ";
620 PRINT "                                              ";
630 IF H = -999 THEN 1080
640      '
650      '*************** Przelicz współrzędne punktów ****************
660 FOR K = 1 TO N
670   RT(K) = R(K) + INT(V / 12)
680   IF RT(K) < 1 OR RT(K) > 15 THEN 980
690   CT(K) = C(K) + INT(H / 8)
700   IF CT(K) < 0 OR CT(K) + LEN(L$(K)) > 35 THEN 980
710   FOR J = 1 TO 2
720     XT(K,J) = X(K,J) + H
730     IF XT(K,J) < 0 OR XT(K,J) > 279 THEN 1000
740     YT(K,J) = Y(K,J) + V
750     IF YT(K,J) < 0 OR YT(K,J) > 159 THEN 1000
760   NEXT J
770 NEXT K
780 RT(N+1) = R(K) + INT(V / 12)
790 CT(N+1) = C(K) + INT(H / 8)
800 GOSUB 830
810 GOTO 520
820 '
830 '******************** Program rysowania ****************************
840 CLEARSCREEN
850 POSITION RT(N+1),CT(N+1)
860 PRINT L$(N+1)
870 PRINT
880 FOR K = 1 TO N
890   POSITION RT(K),CT(K)
900   PRINT L$(K);
910   'Rysuj słupek dla danej części
920   FOR YT = YT(K,1) TO YT(K,2)
930     DRAWLINE XT(K,1),YT TO XT(K,2),YT
940   NEXT YT
950 NEXT K
960 RETURN
970 '*******************************************************************
980 PRINT "Napis poza ekranem"
990 GOTO 1080
```

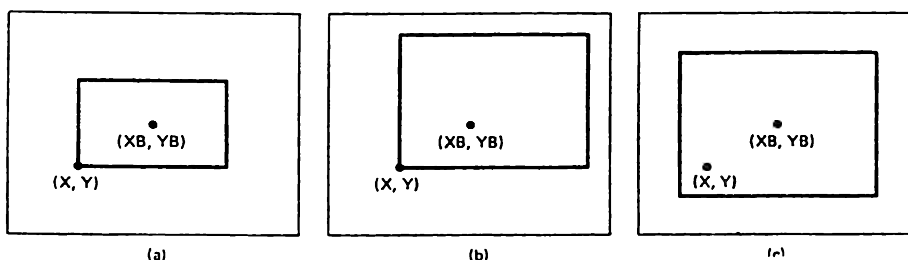
Program 6.2 (cd.)

```
1000 PRINT "Punkt wykresu poza ekranem"
1010 GOTO 1080
1020 '*********************************
1030 DATA 0,800
1040 DATA 4
1050 DATA PIERWSZY,600,DRUGI,500,TRZECI,800,CZWARTY,400
1060 DATA "KWARTALNA SPRZEDAZ"
1070 '*********************************
1080 END
```

6.2. Zmiana wymiarów (skalowanie)

Po utworzeniu obrazu możemy podjąć decyzję o zmianie jego wymiarów, aby polepszyć jakość prezentacji informacji albo dopasować go do większej całości. Wielkość obrazu lub wykresu zmienia się, mnożąc wszystkie odległości między punktami przez współczynnik skali. W celu podwojenia wielkości prostokąta wszystkie jego wymiary liniowe mnoży się przez 2, a w celu zmniejszenia go o połowę wszystkie wymiary mnoży się przez 1/2.

Zmieniając wymiary obiektu musimy również określić położenie obiektu po zwiększeniu lub zmniejszeniu. Skalowania obiektu dokonuje się względem pewnego punktu centralnego obiektu, względem punktu na jego brzegu albo względem jakiegoś punktu zewnętrznego. Na przykład, możemy zdecydować się na powiększenie prostokąta zachowując pozycję dolnego lewego rogu; możemy też powiększyć prostokąt względem jego punktu środkowego (rys. 6.2).



Rys. 6.2. Prostokąt o środku w punkcie (XB, YB) i dolnym lewym rogu w punkcie (X, Y) (a); przy ustalonym punkcie (X, Y) prostokąt powiększa się w prawo i do góry ekranu (b); przy stałym (XB, YB) prostokąt powiększa się od środka na zewnątrz (c)

Skalowanie odcinka

Odcinek poziomy o współrzędnych końców X(1) i X(2) ma długość

$$L = X(2) - X(1) \tag{6.2}$$

Zmiana długości odcinka oznacza mnożenie L przez *współczynnik skali w poziomie* HS; po pomnożeniu otrzymuje się nową długość

$$LS = L * HS = X(2) * HS - X(1) * HS = XS(2) - XS(1) \quad (6.3)$$

przy czym współrzędne przeskalowane są pamiętane w tablicy XS, a współczynnik skali HS jest liczbą dodatnią. Jeżeli HS > 1, to długość odcinka ulegnie zwiększeniu. Krótszy odcinek otrzyma się, gdy HS < 1. Dla HS = 1 długość odcinka nie zmieni się.

Ponieważ skalowanie wpływa na wartość współrzędnych końców odcinka, musimy określić, gdzie ten odcinek ma być narysowany. Dla odcinka poziomego możemy zachować pozycję lewego lub prawego końca, środka odcinka lub możemy przesunąć odcinek do nowego miejsca. Załóżmy, że chcemy zachować lewy koniec odcinka tam, gdzie jest. Wtedy nowe współrzędne odcinka po skalowaniu będą następujące:

$$\begin{aligned} XS(1) &= X(1) \\ XS(2) &= X(1) + (X(2) - X(1)) * HS \end{aligned} \quad (6.4)$$

Jeżeli chcemy zachować prawy koniec odcinka, nowe współrzędne przyjmą postać:

$$\begin{aligned} XS(1) &= X(2) - (X(2) - X(1)) * HS \\ XS(2) &= X2 \end{aligned} \quad (6.5)$$

Zachowując punkt środkowy odcinka otrzymamy:

$$\begin{aligned} XS(1) &= XM - ((X(2) - X(1))/2) * HS \\ XS(2) &= XM + ((X(2) - X(1))/2) * HS \end{aligned} \quad (6.6)$$

przy czym XM jest środkiem odcinka (rys. 6.3) o współrzędnej XM = (X(1)+X(2))/2.

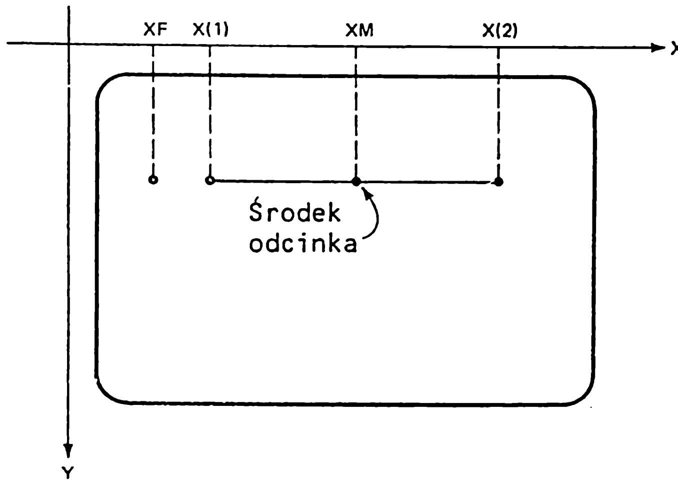
Jeżeli chcemy przesunąć przeskalowany odcinek do nowego miejsca na ekranie, to najpierw przesuwamy odcinek do nowego miejsca, a potem zmieniamy jego wymiary zachowując pozycję jednego z jego końców lub środka. Ogólne podejście polega na wybraniu punktu odniesienia, który uznamy za ustalony (na przykład punkt XF na rys. 6.3). Wtedy nowe końce odcinka otrzymamy, skalując odległości między każdym oryginalnym punktem końcowym a XF. Po skalowaniu nowe współrzędne końców obliczamy jako:

$$\begin{aligned} XS(1) &= (X(1) - XF) * HS + XF \\ XS(2) &= (X(2) - XF) * HS + XF \end{aligned} \quad (6.7)$$

Jako punkt XF można wybrać dowolny punkt ekranu lub spoza ekranu. Przyjmując, że XF jest jednym z punktów X(1), X(2) lub XM otrzymujemy wyniki zgodne odpowiednio z (6.4), (6.5) lub (6.6).

Podobne rozważania są słuszne przy skalowaniu w pionie. Długość odcinka pionowego o końcach Y(1) i Y(2) wynosi

$$L = Y(2) - Y(1) \quad (6.8)$$



Rys. 6.3. Odcinek poziomy o końcach w X(1) i X(2), długości X(2) – X(1) i środku w punkcie o współrzędnej XM = X(1) + (X(2) – X(1))/2

Skalowanie takiego odcinka za pomocą *współczynnika skali w pionie* VS da odcinek o nowej długości

$$LS = L * VS = Y(2) * VS - Y(1) * VS = YS(2) - YS(1) \tag{6.9}$$

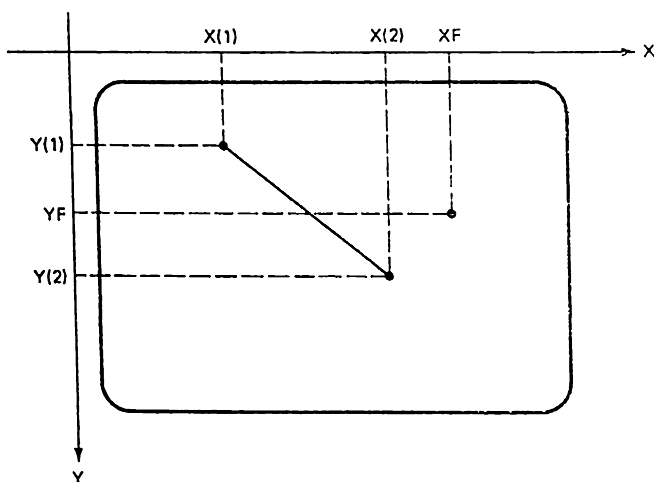
przy czym nowe współrzędne końców są pamiętane w tablicy YS. Współczynnik skali VS musi być dodatni i odcinek można narysować względem dowolnego ustalonego punktu, na przykład Y(1), Y(2) lub środka odcinka. Oznaczając pionowy punkt ustalony przez YF, możemy wyznaczyć przeskalowane współrzędne jako:

$$\begin{aligned} YS(1) &= (Y(1) - YF) * VS + YF \\ YS(2) &= (Y(2) - YF) * VS + YF \end{aligned} \tag{6.10}$$

Wyświetlane odcinki ukośne (rys. 6.4) można powiększać lub zmniejszać skalując obie składowe długości: poziomą X(2) – X(1) oraz pionową Y(2) – Y(1) względem *ustalonego punktu* (XF, YF). Zależności (6.7) i (6.10) możemy napisać w następującej postaci:

$$\begin{aligned} XS(1) &= X(1) * HS + XF * (1 - HS) \\ YS(1) &= Y(1) * VS + YF * (1 - VS) \\ XS(2) &= X(2) * HS + XF * (1 - HS) \\ YS(2) &= Y(2) * VS + YF * (1 - VS) \end{aligned} \tag{6.11}$$

Wartości wyrażeń XF * (1 – HS) i YF * (1 – VS) są stałe i musimy je wyznaczać tylko raz dla wszystkich współrzędnych. Chcąc zwiększyć lub zmniejszyć odcinek leżący wzdłuż przekątnej, bez zmiany nachylenia, musimy przyjąć HS = VS.



Rys. 6.4. Odcinek ukośny o poziomej składowej długości X(2) – X(1) i pionowej Y(2) – Y(1) skalowany względem dowolnego ustalonego punktu (XF, YF)

Skalowanie obrazu

Jednolite zwiększenie lub zmniejszenie wymiarów obrazu lub wykresu uzyskuje się, przyjmując taką samą wartość HS i VS (równą S). Korzysta się wtedy z zależności (6.11) do obliczania nowych współrzędnych dla wszystkich punktów skalowanego obrazu. Zwiększony lub zmniejszony obraz uzyskuje się, łącząc odcinkami nowo otrzymane punkty. Obliczanie współrzędnych każdego punktu obrazu za pomocą (6.11) nie jest konieczne w przypadku figur ograniczonych liniami krzywymi. Na przykład, aby zmienić wymiary okręgu o promieniu R, wystarczy obliczyć za pomocą (6.11) nowe współrzędne środka koła oraz nowy promień jako R * S. Z kolei, do wyświetlenia przeskalowanego okręgu można korzystać z programów z p. 5.1.

Niekiedy zależy nam na rozciągnięciu wymiarów obrazu w jednym kierunku (np. w pionie) i zachowaniu oryginalnych wymiarów w drugim kierunku (w poziomie). Aby zmienić kwadrat w prostokąt, wystarczy tylko zwiększyć wysokość. W tym celu do obliczenia pionowych współrzędnych punktów przyjmujemy na przykład VS = 2, a dla poziomych HS = 1. Sposób ten może być użyteczny przy eksperymentowaniu lub dobieraniu współrzędnych elementów obrazu. Niejednolite skalowanie możemy również stosować do konstruowania złożonego obrazu za pomocą zbioru standardowych kształtów: na przykład kwadrat po skalowaniu może stać się prostokątem o żądanych wymiarach.

Przykład skalowania obrazu przedstawiono w programie 6.3. Danymi wejściowymi do programu są definicja obrazu oraz współczynniki skali HS i VS. Pro-

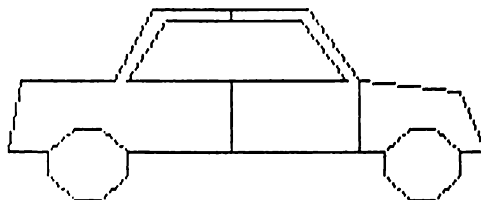
gram dokonuje skalowania całego obrazu lub dowolnego wskazanego elementu obrazu. Przykładowy wynik pokazano na rys. 6.5.

Program 6.3. Skalowanie części obrazu (samochód)

```
10  'PROGRAM 6.3. Skalowanie całości lub części obrazu
20      'Program rysuje samochód i umożliwia skalowanie całości lub
30      'części rysunku. Przy skalowaniu części program określa punkt
40      'stały. Przy skalowaniu całości użytkownik może wprowadzić punkt
50      'stały. Punkty otrzymane po skalowaniu zastępują oryginalne
60      'wartości w tablicach X i Y.
70  CLEARSCREEN
80  DIM X(5,100), Y(5,100), N(5)
90  PRINT "Podaj wartości maksymalne poziomą i pionową"
100 PRINT "dla przyjętej rozdzielczości"
110 INPUT XM, YM
120     '
130     '************** Odczytaj części obrazu i rysuj ****************
140 PN = 1      'PN - numer części
150 E = 0       'E - element części
160 READ XD, YD
170 IF XD >= 0 THEN 240          'Czy to jest nowy punkt? Jeżeli nie,
180     'to koniec części. Zapamiętaj bieżącą wartość E w N(PN)
190 N(PN) = E
200 IF XD = -100 THEN 280        'Jeżeli koniec danych,to rysuj obraz.
210     'Jeżeli nie,to nowa część. Wczytaj elementy tej części
220 PN = PN + 1
230 GOTO 150
240 E = E + 1
250 X(PN,E) = XD
260 Y(PN,E) = YD * 5/6     'Korekta współrzędnej Y ze względu na roz-
270 GOTO 160                                               'dzielczość
280 GRAPHICS
290 GOSUB 770
300     '
310     '****************** Instrukcje drukowania ******************
320 POSITION 21,1
330 PRINT "1-Przód  2-Część środkowa  3-Tył  4-Całość  0-Stop";
340 POSITION 22,1
350 PRINT "Którą część skalować?";
360 INPUT P
370 POSITION 21,1
380 PRINT "                                               ";
390 PRINT "                                               ";
400 IF P = 0 THEN 1100
410 POSITION 21,1
420 PRINT "Podaj wartości współczynników skali HS i VS"
430 INPUT HS, VS
440 POSITION 21,1
450 PRINT "                                               ";
460     '
470     '******** Przelicz punkty odpowiedniej części obrazu ********
480     'Przy skalowaniu części obrazu , skalowanie jest względem
490     'punktu centralnego samochodu. Przy skalowaniu całego samochodu
500     'użytkownik musi wprowadzić punkt stały
510 IF P = 4 THEN 670          'Skalowanie wszystkich części
520     'Skalowanie jednej części
530 IF P = 1 THEN 560
540 IF P = 2 THEN 590
550 IF P = 3 THEN 620
560 XF = X(1,2)
570 YF = Y(1,2)
580 GOTO 640
```

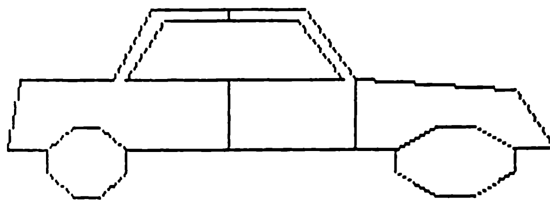
Program 6.3 (cd.)

```
590 XF = X(2,2)
600 YF = Y(2,2)
610 GOTO 640
620 XF = X(3,2)
630 YF = Y(3,2)
640 GOSUB 870              'Skaluj
650 GOSUB 770              'Rysuj
660 GOTO 320
670 POSITION 21,1          'Skalowanie całego samochodu
680 PRINT "Podaj współrzędne punktu stałego";
690 INPUT XF, YF
700 FOR P = 1 TO PN        'Skaluj poszczególne części
710   GOSUB 870
720 NEXT P
730 GOSUB 770              'Rysuj
740 GOTO 320
750 '
760 '******************** Program rysowania ****************************
770      'Rysowanie obrazu
780 CLEARSCREEN
790 FOR K = 1 TO PN
800   FOR J = 1 TO N(K)-1
810     IF X(K,J)<0 OR X(K,J)>XM OR Y(K,J)<0 OR Y(K,J)>YM THEN 1090
820     DRAWLINE X(K,J), Y(K,J) TO X(K,J+1),Y(K,J+1)
830   NEXT J
840 NEXT K
850 RETURN
860 '******************** Program skalowania **************************
870      'Skaluj części obrazu
880 FOR J = 1 TO N(P)
890   X(P,J) = X(P,J) * HS + XF * (1 - HS)
900   Y(P,J) = Y(P,J) * VS + YF * (1 - VS)
910 NEXT J
920 RETURN
930 '*****************************************************************
940      'Przód samochodu
950 DATA 220,90,210,90,210,60,250,65,260,90,250,90
960 DATA 250,100,240,110,230,110,220,100,220,90
970 DATA 230,80,240,80,250,90
980 DATA -1,-1
990      'Część środkowa
1000 DATA 210,90,160,90,160,60,205,60,187,35,160,35
1010 DATA 160,30,190,30,210,60
1020 DATA -1,-1
1030     'Tył samochodu
1040 DATA 120,90,160,90,160,60,120,60,135,35,160,35
1050 DATA 160,30,130,30,115,60,80,60,75,90,90,90
1060 DATA 90,100,100,110,110,110,120,100,120,90,110,80,100,80,90,90,90,100
1070 DATA -100,-100
1080 '*****************************************************************
1090 PRINT "Współrzędna poza zakresem"
1100 END
```



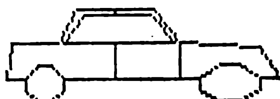
1-Przód 2-Część środkowa 3-Tył 4-Całość
0-Stop
Którą część skalować?

(a)



1-Przód 2-Część środkowa 3-Tył 4-Całość
0-Stop
Którą część skalować?

(b)



1-Przód 2-Część środkowa 3-Tył 4-Całość
0-Stop
Którą część skalować?

(c)

Rys. 6.5. Skalowanie części pierwotnego obrazu (a) za pomocą programu 6.3; powiększenie części przedniej (b); zmniejszenie całości (c)

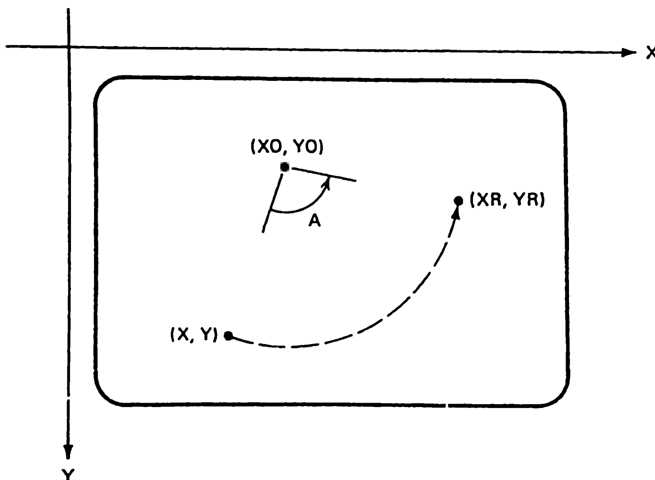
Z metod stosowanych w programie 6.3 korzysta się przy skalowaniu wykresów. Ponieważ opisów nie można skalować (wielkość znaków jest ustalona), musimy rozważyć rozmieszczenie napisów po skalowaniu na poziomie pikseli. Możemy to zrobić analizując wykres po skalowaniu albo możemy obliczyć nowe pozycje początkowe dla łańcuchów znaków względem pewnego punktu wykresu. Alternatywnie moglibyśmy tworzyć opisy z pikseli. Wtedy opisy mogłyby być skalowane razem z innymi częściami wykresu.

6.3. Zmiana orientacji (obroty)

W wielu zastosowaniach zachodzi potrzeba zmiany orientacji obrazu. Możemy zdecydować się na taką zmianę wykresu słupkowego, aby słupki były rysowane poziomo, a nie pionowo. Można zrealizować to bez konieczności rekonstruowania wykresu, a jedynie obracając obraz o 90°. W zastosowaniach związanych z symulacją i grami często wyświetla się obracające się obrazy.

Obrót punktu

Na rysunku 6.6 przedstawiono sposób obracania punktu. Obracany punkt przesuwa się wzdłuż łuku z pozycji (X, Y) do (XR, YR) wokół środka obrotu (X0, Y0). Kąt A określa wielkość obrotu od punktu (X, Y) do (XR, YR). Współ-



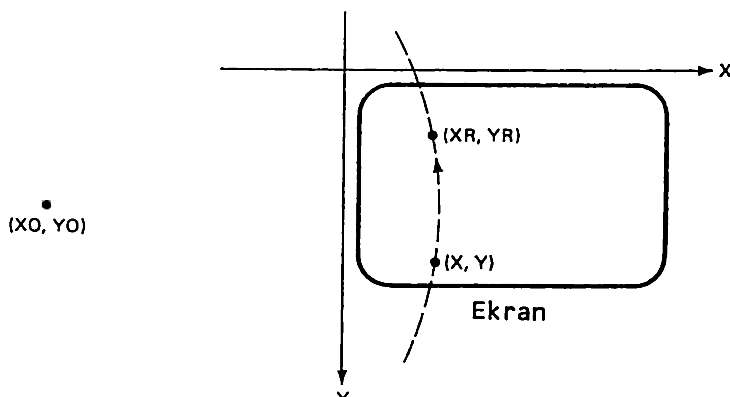
Rys. 6.6. Obrót punktu z pozycji (X, Y) do (XR, YR) wokół środka obrotu (X0, Y0). Punkt obraca się wzdłuż okręgu o kąt A

rzędne punktu po obrocie oblicza się przy znanych wartościach: A, (X0, Y0) i (X, Y) z zależności:

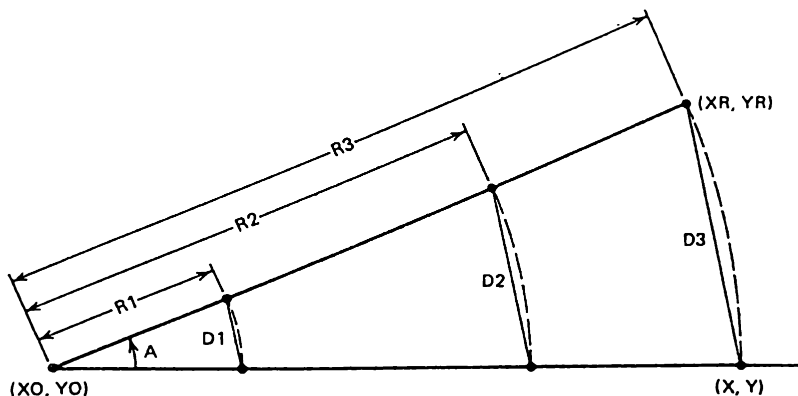
$$XR = X0 + ((X - X0) * COS(A)) + ((Y - Y0) * SIN(A))$$
$$YR = Y0 + ((Y - Y0) * COS(A)) - ((X - X0) * SIN(A)) \quad (6.12)$$

Środek obrotu (X0, Y0) można wybrać w dowolnym miejscu – albo na ekranie, albo poza nim. Ten punkt nie jest wyświetlany; wybiera się go jedynie jako punkt odniesienia w celu określenia toru obrotu (rys. 6.7).

Przy założeniu, że początek układu współrzędnych znajduje się w górnym lewym rogu obrazu, kąt obrotu A mierzy się w kierunku przeciwnym względem ruchu wskazówek zegara zaczynając od punktu początkowego (X, Y). W systemach



Rys. 6.7. Środek obrotu (X0, Y0) można wybrać na ekranie lub poza nim



Rys. 6.8. Dla małych wartości kąta obrotu A odległość od punktu (X, Y) do (XR, YR) jest w przybliżeniu równa iloczynowi promienia R przez kąt obrotu A. Dla danego kąta A większemu R odpowiada większe D

komputerowych, w których początek układu współrzędnych znajduje się w dolnym lewym rogu, kąt obrotu A jest mierzony w kierunku zgodnym z ruchem wskazówek zegara. Kąt ten ma zazwyczaj wartość w przedziale od 0 do 2 * PI (6, 283185) radianów. Można określać inne kąty, ale one po prostu powtarzają obroty w tym przedziale.

Odległość, o jaką przesuwa się punkt wzdłuż okręgu dla danego kąta A, zależy od odległości punktu (X, Y) od środka obrotu (X0, Y0). Im (X, Y) jest dalej od środka obrotu, tym większa jest droga przebyta od (X, Y) do (XR, YR). Zależności między odległością R od środka obrotu, kątem A i przesunięciem D od (X, Y) do (XR, YR) pokazano na rys. 6.8. Dla małych kątów obrotu, cięciwa D jest w przybliżeniu równa iloczynowi R * A. Dla R = 50 punkt można obrócić o ok. 1 odległość jednostkową (do sąsiedniego piksela) przy kącie A = 0,02 rad.

Obrót obrazu

W celu zmiany orientacji wyświetlonego obrazu najpierw wybiera się środek obrotu. Następnie z zależności (6.12) oblicza się współrzędne wszystkich obróconych punktów. Wreszcie zeruje się ekran i na nowo rysuje linie obiektu korzystając ze współrzędnych obróconych punktów.

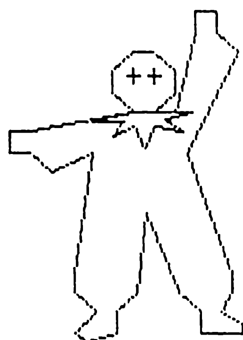
Program 6.4 obraca obraz wokół wybranego środka obrotu (X0, Y0) o określony kąt obrotu A przy założeniu, że obrócony obraz znajduje się na ekranie. Wyniki działania programu dla kątów obrotu 90° i 180° pokazano na rys. 6.9. W tym przykładzie środek obrotu wybrano w środku obrazu.

Program 6.4. Obracanie obrazów (klown)

```
10  'PROGRAM 6.4.Obracanie obrazu
20      'Program rysuje obraz. Użytkownik wprowadza kąt obrotu. Obróco-
30      'ne punkty są pamiętane w tablicach XR i YR.
40  CLEARSCREEN
50  DIM X(7,100), Y(7,100), N(7)
60  DIM XR(7,100), YR(7,100)
70  PRINT "Podaj wartości maksymalne poziomą i pionową"
80  PRINT "dla przyjętej rozdzielczości"
90  INPUT XM, YM
100     '
110     '************** Odczytaj części obrazu i rysuj ****************
120 PN = 1                  'PN - numer części
130 E = 0                   'E - element części
140 READ XD, YD
150 IF XD >= 0 THEN 220          'Czy to jest nowy punkt? Jeżeli nie,
160     'to koniec części. Zapamiętaj bieżącą wartość E w N(PN)
170 N(PN) = E
180 IF XD = -100 THEN 260        'Jeżeli koniec danych,to rysuj obraz
190     'Jeżeli nie,to nowa część. Wczytaj elementy tej części
200 PN = PN + 1
210 GOTO 130
220 E = E + 1
230 X(PN,E) = XD
240 Y(PN,E) = YD * 5/6    'Korekta współrzędnych Y punktów ze względu na
250 GOTO 140                                             'rozdzielczość
260     'Kopiuj wartości początkowe do tablic przeznaczonych do pamię-
270     'tania współrzędnych obróconych punktów. Program rysowania
275     'korzysta z tych tablic
```

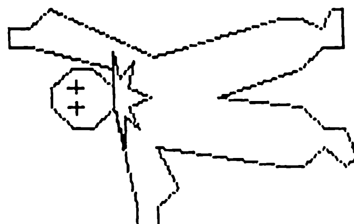
Program 6.4 (cd.)

```
280 FOR P = 1 TO PN
290   FOR E = 1 TO N(P)
300     XR(P,E) = X(P,E)
310     YR(P,E) = Y(P,E)
320   NEXT E
330 NEXT P
340 GRAPHICS
350 GOSUB 590                      'Rysuj obraz
360      '
370      '****************** Instrukcje drukowania ******************
380 POSITION 22,1
390 PRINT "Podaj kąt obrotu względem oryginalnego położenia";
400 POSITION 23,1
410 PRINT "(w zakresie od 0 do 360 stopni lub napisz -1,aby zakończyć";
415 PRINT "program)";
420 INPUT A
430 POSITION 22,1
440 PRINT "                                                          ";
450 PRINT "                                                          ";
460 IF A < 0 THEN 900
470 A = A * 3.14159 / 180       'Zamiana stopni    radiany
480 POSITION 22,1
490 PRINT "Wokół jakiego punktu obrócić?";
500 INPUT XO, YO
510      '
520      '********************* Przelicz punkty *********************
530 FOR P = 1 TO PN                'Obróć każdą część
540   GOSUB 680
550 NEXT P
560 GOSUB 590                      'Rysuj
570 GOTO 370
580 '
590 '******************** Program rysowania ***************************
600 CLEARSCREEN
610 FOR P = 1 TO PN
620   FOR E = 1 TO N(P)-1
630     IF X(P,E)<0 OR X(P,E)>XM OR YR(P,E)<0 OR YR(P,E)>YM THEN 890
640     DRAWLINE XR(P,E), YR(P,E) TO XR(P,E+1), YR(P,E+1)
650   NEXT E
660 NEXT P
670 RETURN
680 '******************** Program obracania ***************************
690 FOR E = 1 TO N(P)
700   XR(P,E) = XO + (X(P,E)-XO) * COS(A) + (Y(P,E)-YO) * SIN(A) * 6/5
710   YR(P,E) = YO + (Y(P,E)-YO) * COS(A) - (X(P,E)-XO) * SIN(A) * 5/6
720 NEXT E
730 RETURN
740 '***************************************************************
750 DATA 160,60,170,50,170,40,160,30,150,30,140,40,140,50
760 DATA 150.60,178,60,165,63,173,70,160.67,155,78,150,67
770 DATA 140,72,145,65,130,65,150,60,100,70,90,70,90,80,100
780 DATA 80,110,90,130,80,120,140,120,150,130,160,120,170,125.
790 DATA 175,140,170,140,160,150,150,155,110,170,150,180,160,180
800 DATA 170,200,170,200,165,190,160,195,150,195,140,175,80,200
810 DATA 30,190,20,190,10,180,10,180,20,170,60
820 DATA -1,-1
830 DATA 147,42,153,42,-1,-1
840 DATA 150,38,150,45,-1,-1
850 DATA 157,42,163,42,-1,-1
860 DATA 160,38,160,45
870 DATA -100,-100
880
'******************************************************************
890 PRINT "Współrzędna poza zakresem"
900 END
```



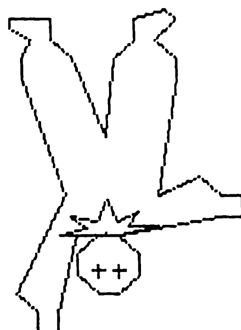
Podaj kąt obrotu względem oryginalnego położenia
(w zakresie od 0 do 360 stopni lub napisz -1 aby
zakończyć program)

(a)



Podaj kąt obrotu względem oryginalnego położenia
(w zakresie od 0 do 360 stopni lub napisz -1 aby
zakończyć program)

(b)



Podaj kąt obrotu względem oryginalnego położenia
(w zakresie od 0 do 360 stopni lub napisz -1 aby
zakończyć program)

Rys. 6.9. Obracanie obrazu za pomocą programu 6.4: a) pozycja początkowa; b) obrót o 90°; c) obrót o 180°

Obrót może zniekształcić obraz lub wykres, jeżeli rozdzielczości X i Y w systemie są różne. W programie 6.4 uwzględniamy ten efekt mnożąc odpowiednie czynniki przez stosunek rozdzielczości X do Y w wierszu 700 oraz przez stosunek rozdzielczości Y do X w wierszu 710. Jest to równoważne obracaniu obrazu wzdłuż elipsy, która na ekranie wygląda jak okrąg.

6.4. Łączenie transformacji

Program 6.5 umożliwia łączenie transformacji: przesuwania, skalowania i obrotu. Pokazuje procedurę konstruowania wykresu słupkowego za pomocą transformacji. Wykres tworzymy w górnym lewym rogu ekranu, następnie kolejno wybieramy transformacje lub kod kończący program. Umożliwia to sprawdzanie różnych położeń wykresu na ekranie i dobór wymiarów obrazu aż do uzyskania zadowalającego efektu końcowego. Na rysunku 6.10 przedstawiono oryginalny wykres i jego końcowe położenie na ekranie.

Program 6.5. Łączenie transformacji na przykładzie wykresu

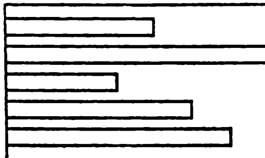
```
10  'PROGRAM 6.5. Transformacja wykresu słupkowego
20       'Założone wymiary ekranu 280 x 160 pikseli. Program wczytuje
30       'zmienną liczbę wartości danych (do 8) z instrukcji DATA i rysu-
40       'je oś X,a wartości danych są przeliczane tak,by zawierały
50       'się w przedziale 10-110 pikseli. Wierzchołki słupków dla
60       'każdego wykresu są pamiętane w tablicach X i Y. Po utworze-
70       'niu wykresu można użyć dowolnej kombinacji transformacji
80       'przesunięcia, skalowania i obrotu. Nowe punkty zastępują
90       'wartości oryginalne w tablicach X i Y.
100      '
110 CLEARSCREEN
120 DIM X(8,4), Y(8,4), N(8)
130      '
140      '****************** Odczytaj dane dla wykresu ******************
150 READ MI, MA     'Odczytaj wartości najmniejszą i największą dla
160 RM = (110 - 10) / (MA - MI)              'przyjętego zakresu danych
170 READ D
180 IF D <= 8 THEN 210
190 PRINT "Dopuszcza się tylko 8 części"
200 GOTO 1200
210 FOR K = 1 TO D
220   READ M
230   N(K) = 4
240   X(K,1) = 10
250   Y(K,1) = 15 + (K-1) * 10    'Kolejny słupek jest rysowany poniżej
260   X(K,2) = INT((M - MI) * RM + 10 + 0.5)                'poprzedniego
270   Y(K,2) = Y(K,1)
280   X(K,3) = X(K,2)
290   Y(K,3) = Y(K,1) + 6
300   X(K,4) = 10
310   Y(K,4) = Y(K,3)
320 NEXT K
330      'Zapamiętaj punkty do rysowania osi
340 K = D + 1
350 N(K) = 3
360 X(K,1) = 110
```

Program 6.5 (cd.)

```
370 Y(K,1) = 10
380 X(K,2) = 10
390 Y(K,2) '= 10
400 X(K,3) = 10
410 Y(K,3) = 15 +10 * D
420 GRAPHICS   'Rysuj wykres
430 GOSUB 610
440      '
450      '***************** Instrukcje drukowania *********************
460 POSITION 22,1
470 PRINT "1-Przesunięcie  2-Skalowanie  3-Obrót  4-Stop";
480 POSITION 23,1
490 PRINT "Jaka transformacja?";
500 INPUT T
510 POSITION 22,1
520 PRINT "                                                  "
530 PRINT "                                                  ";
540      'Przejście do programu określonej transformacji
550 IF T =1 THEN GOSUB 710
560 IF T =2 THEN GOSUB 820
570 IF T =3 THEN GOSUB 1010
580 IF T =4 THEN 1220
590 GOSUB 610  'Rysuj
600 GOTO 450
610 '******************* Program rysowania ***************************
620 CLEARSCREEN
630 FOR K = 1 TO D+1
640   FOR J = 1 TO N(K)-1
650     IF X(K,J)<0 OR X(K,J)>279 OR X(K,J+1)<0 OR X(K,J+1)>279 THEN 1200
660     IF Y(K,J)<0 OR Y(K,J)>159 OR Y(K,J+1)<0 OR Y(K,J+1)>159 THEN 1200
670     DRAWLINE X(K,J),Y(K,J) TO X(K,J+1),Y(K,J+1)
680   NEXT J
690 NEXT K
700 RETURN
710 '******************* Program przesunięcia *************************
720 POSITION 22,1
730 PRINT "Podaj wartości przesunięcia H i V";
740 INPUT H, V
750 FOR K = 1 TO D + 1
760   FOR J = 1 TO N(K)
770     X(K,J) = X(K,J) + H
780     Y(K,J) = Y(K,J) + V
790   NEXT J
800 NEXT K
810 RETURN
820 '******************* Program skalowania ****************************
830 POSITION 22,1
840 PRINT "Podaj wartości współczynników skali HS i VS";
850 INPUT HS, VS
860 POSITION 22,1
870 PRINT "                                            "
880 POSITION 22,1
890 PRINT "Podaj współrzędne punktu stałego";
900 INPUT XF, YF
910      'Oblicz część stałą w równaniu
920 XC = XF * (1-HS)
930 YC = YF * (1-VS)
940 FOR K = 1 TO D + 1
950   FOR J = 1 TO N(K)
960     X(K,J) = X(K,J) * HS + XC
970     Y(K,J) = Y(K,J) * VS + YC
980   NEXT J
```

Program 6.5 (cd.)

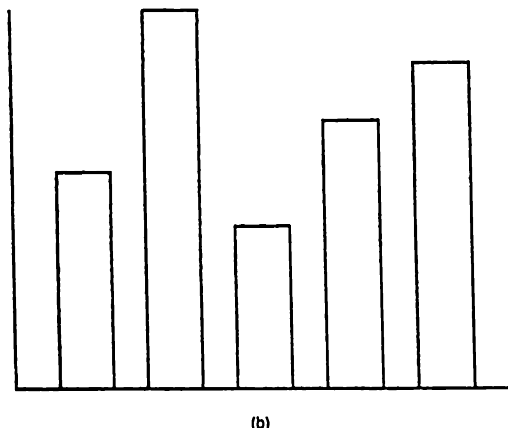
```
990 NEXT K
1000 RETURN
1010 "****************** Program obrotu ********************************
1020 POSITION 22,1
1030 PRINT "Podaj wartość kąta obrotu";
1040 INPUT A
1050 A = A * 3.14159 / 180              'Zamiana na radiany
1060 POSITION 22,1
1070 PRINT "                                             "
1080 POSITION 22,1
1090 PRINT "Wokół którego punktu obrócić?";
1100 INPUT XO, YO
1110 FOR K = 1 TO D + 1
1120   FOR J = 1 TO N(K)
1130     XH(K,J) = X(K,J)     'Zachowujemy X(K,J) do obliczania Y
1140     X(K,J) = XO + (X(K,J)-XO) * COS(A) + (Y(K,J)-YO) * SIN(A) * 6/5
1150     Y(K,J) = YO + (Y(K,J)-YO) * COS(A) - (XH(K,J)-XO) * SIN(A) * 5/6
1160   NEXT J
1170 NEXT K
1180 RETURN
1190 '*******************************************************************
1200 PRINT "Współrzędna poza zakresem"
1210 DATA 0,700,5,400,700,300,500,600
1220 END
```



1-Przesunięcie 2-Skalowanie 3-Obrót 4-Stop
Jaka transformacja?

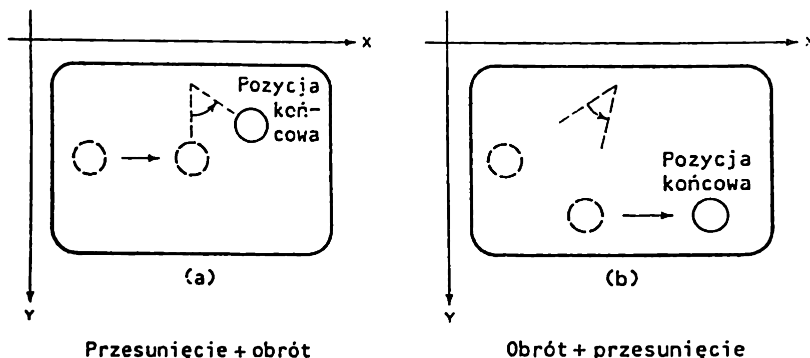
(a)

Rys. 6.10a (podpis na stronie 130)



Rys. 6.10. Wykres w początkowej pozycji (a); wykres uzyskany za pomocą programu 6.5, w którym wykorzystano transformacje: przesunięcia, skalowania i obrotu (b)

Kolejność, w jakiej wykonujemy przesunięcie i obrót, może mieć wpływ na końcową pozycję wyświetlanego obiektu. Jeżeli obracamy obiekt wokół punktu zewnętrznego w stosunku do obiektu, to końcowa pozycja po transformacjach zależy od tego, czy dokonujemy przesunięcia przed czy po obrocie (rys. 6.11).



Rys. 6.11. Końcowa pozycja obiektu po przesunięciu i obrocie (wokół punktu zewnętrznego względem obiektu) zależy od kolejności tych transformacji: a) najpierw wykonano przesunięcie; b) przesunięcie wykonano na końcu

ZADANIA

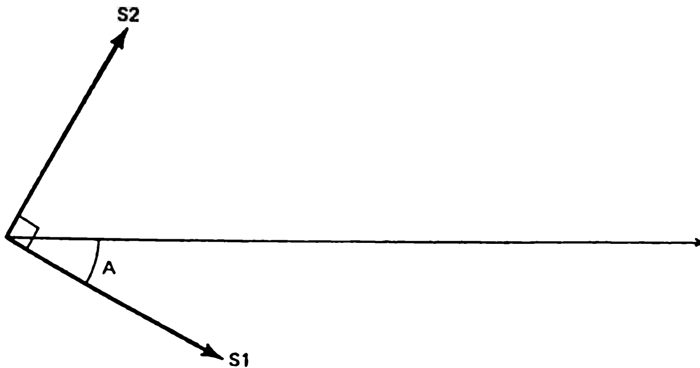
6.1. Napisz program przesunięcia okręgu do zadanego miejsca na ekranie, obliczając nowy środek okręgu i ponownie rysując okrąg w nowym miejscu.

6.2. Zmodyfikuj program z zadania 6.1 tak, aby można było zmieniać wielkość okręgu (względem jego środka) i zamaluj jego wnętrze po przesunięciu. Parametry dla przesuwania, skalowania i kolorowania koła są określane w danych wejściowych.

6.3. Napisz program przesuwania ciągu znaków z jednego miejsca ekranu w inne, korzystając z metody stosowanej w programie 6.2. Jako dane wejściowe podaj wymiary siatki znaków w pikselach i wielkość przesunięcia. Żadna część ciągu znaków nie może być przesunięta poza ekran.

6.4. Napisz program skalowania i wyświetlania dowolnego wielokąta określonego w danych wejściowych. Przyjmij, że punktem stałym jest środek wielokąta (środek ciężkości). Współrzędną X środka ciężkości oblicz jako wartość średnią współrzędnych X wierzchołków, a współrzędną Y środka ciężkości jako wartość średnią współrzędnych Y wierzchołków.

6.5. Napisz program skalowania zadanego obrazu w wybranym kierunku. Kierunek ma być określony przez kąt A mierzony względem poziomu (rys. 6.12). Takie skalowanie można otrzymać obracając obiekt w kierunku przeciwnym względem ruchu wskazówek zegara o kąt A, dokonując skalowania przyjmując HS = S1 i VS = S2 i obracając obiekt z powrotem (obrót o kąt −A) do początkowej pozycji. Parametry A, S1 i S2 mają być określone jako dane wejściowe.



Rys. 6.12. Kierunki skalowania określone przez kąt A dla współczynników skali S1 i S2

6.6. Zmodyfikuj program 6.5 tak, aby można było automatycznie zmieniać położenie napisów przy zmianie wymiarów wykresu. Nową początkową pozycję opisu określ skalując odległość od ustalonego punktu do oryginalnej pozycji początkowej.

6.7. Napisz program, który będzie wykonywał dowolną transformację (przesunięcie, skalowanie, obrót) słowa złożonego z dużych liter narysowanych za pomocą pikseli. Rozciągnięcie liter w kierunku ukośnym (tak jak w zadaniu 6.5) pochyla litery.

6.8. Napisz ogólny program transformacji, który będzie przesuwał, skalował lub obracał dowolny obraz lub scenę. Obraz lub scenę można zapamiętać w tablicach dwuwymiarowych na podstawie rysunku na papierze milimetrowym. Parametry wejściowe będą określały rodzaj transformacji, którą należy wykonać.

7. Animacja

Z podstawowych metod transformacji (przesunięcie, skalowanie, obrót) możemy korzystać do animacji obrazów. Stosowana tu technika polega na wyświetleniu obrazu, wymazaniu go, dokonaniu transformacji i wyświetleniu obiektu po transformacji. Powtarzając tę procedurę kilka razy otrzymamy ruch. W ten sposób tworzy się filmy animowane. Artysta rysuje kolejne ramki filmu z niewielkimi zmianami położenia i wymiarów obiektu. Oglądając szybko kolejne ramki, uzyskujemy wrażenie ruchu.

7.1. Punkty i okręgi

Programy tworzenia poruszającego się po ekranie piksela są adaptacjami programów przesuwania omówionych w rozdz. 6. Zamiast przesuwać punkt raz, utrzymujemy go w ruchu. Możemy spowodować ruch punktu wzdłuż linii prostej albo krzywej.

Ruch wzdłuż linii prostej

Jako prosty przykład animacji wzdłuż linii prostej rozważmy poziomy ruch piksela w poprzek ekranu. W celu przesunięcia piksela od lewej do prawej strony możemy zacząć od pewnego miejsca, powiedzmy (X1, Y), i skończyć w innym miejscu (X2, Y), przy czym X2 > X1. W takiej sytuacji proces animacji składa się z szeregu kolejnych przesunięć jednostkowych od X1 do X2. W każdym kroku usuwamy punkt (X, Y) narysowany poprzednio i rysujemy następny punkt (X + 1, Y). Ruch w lewo uzyskujemy zmniejszając wartość współrzędnej X o jeden w każdym kroku. Łącząc oba te ruchy możemy odbijać punkt między X1 i X2. Taki ruch realizuje program 7.1. Punkt początkowo porusza się w prawo z dodatnim przyrostem jednostkowym (DX = 1). Po osiągnięciu punktu (X2, Y) zmieniamy wartość przy-

Program 7.1. Odbijanie punktu w poziomie

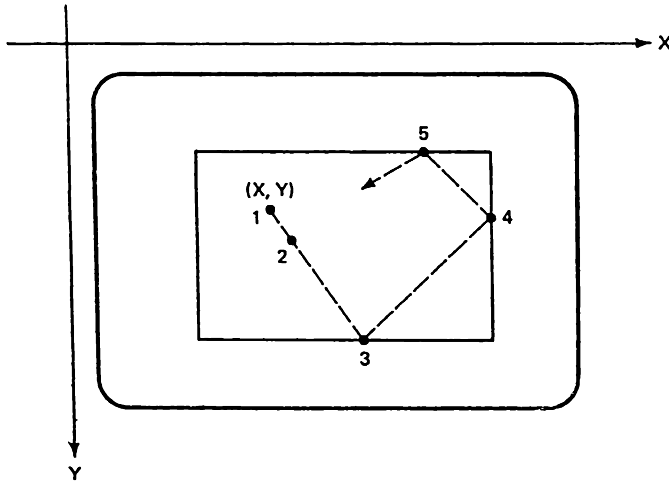
```
10  'PROGRAM 7.1. Odbijanie punktu między pionowymi ścianami
20  CLEARSCREEN
30  PRINT "Podaj wartości maksymalne poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM, YM
60  PRINT "Podaj wartości współrzędnych X dla ścian lewej i prawej"
70  INPUT XL, XR
80  IF XL >= XR OR XL < 0 OR XR > XM THEN 210
90  DX = 1
100 X = XL + INT((XR - XL) / 2)     'Punkt początkowy w środku między
110 Y = INT(YM / 2)                                          'ścianami
120 CLEARSCREEN
130 GRAPHICS
140  '*************************************************************
150 POINTPLOT X,Y
160 IF X > XL AND X < XR THEN 180   'Jeżeli wciąż między ścianami,konty-
170 DX = -DX                        'nuuj. W przeciwnym wypadku zawróć
180 POINTOFF X,Y
190 X = X + DX                     'Przesuń punkt
200 GOTO 150
210 END
```

rostu na ujemną (DX = −DX) i punkt będzie poruszał się w lewo. W punkcie (X1, Y) ponownie zmieniamy znak przyrostu. Program powtarza ruch bez końca.

Program 7.1 można tak zmodyfikować, aby rysował pionowe linie ograniczenia w punktach o współrzędnych X1 i X2. Dzięki temu uzyskuje się wrażenie, że punkt odbija się od ścian. Wtedy jednak należy zmienić punkty zwrotne tak, aby nie wymazywać ściany po odbiciu punktu. Osiągamy to zmieniając kierunek ruchu w punkcie odległym o przyrost jednostkowy od ściany.

Poruszanie punktu tam i z powrotem wzdłuż prostej w kierunku Y oznacza, że ustalamy granice pionowe (Y1 i Y2) i animujemy punkt przy ustalonej współrzędnej X. W celu uzyskania ruchu w dowolnym innym kierunku zwiększamy zarówno współrzędną X, jak i współrzędną Y. Proces ten przypomina kreślenie linii prostej z tym, że teraz usuwamy każdy narysowany punkt, zanim narysujemy następny. Tor ruchu określa się rozmaicie. Możemy wybrać punkty końcowe toru, możemy korzystać z równania prostej (nachylenie i przesunięcie w kierunku Y) albo możemy wybrać dwa przyrosty dla współrzędnych X i Y (DX i DY). W pierwszych dwóch przypadkach musimy tak ustalić wartości DX i DY, aby stosunek DY/DX był równy nachyleniu linii.

Przykład ruchu wzdłuż linii ukośnych podano w programie 7.2. Odbijamy punkt od wewnętrznych ścian pudełka. Wybieramy wymiary pudełka i rozpoczynamy od przesuwania punktu wzdłuż linii ukośnej do dołu i w prawo z przyrostem równym 1 dla obu współrzędnych. Gdy punkt dotrze do ściany pudełka, kierunek ruchu ulega zmianie, tak jak to pokazano na rys. 7.1. Jeżeli punkt odbije się od pionowego boku, zmienia się znak przyrostu X (DX = −DX). Jeżeli punkt odbije się od poziomego boku, zmienia się znak przyrostu dla współrzędnej Y (DY = −DY). W narożniku oba przyrosty zmieniają znak.



Rys. 7.1. Tor piksela odbijanego wewnątrz prostokąta. W punkcie 1 piksel ma współrzędne (X, Y), w punkcie 2 zaś (X + DY, Y + DY). W punkcie 3 kierunek ruchu zmienia się przez zmianę DY na −DY. Kierunek ruchu ulega zmianie ponownie w punkcie 4, w którym przyjmuje się, że DX = −DX, oraz w punkcie 5, w którym DY = −DY

Program 7.2. Odbijanie punktu wewnątrz prostokąta z wykorzystaniem przyrostów jednostkowych

```
10  'PROGRAM 7.2. Odbijanie punktu wewnątrz prostokąta
20       'DX = DY = 1.
30  CLEARSCREEN
40  PRINT "Podaj wartości maksymalne poziomą i pionową"
50  PRINT "dla przyjętej rozdzielczości"
60  INPUT XM, YM
70  PRINT "Podaj wartości współrzędnych X dla lewego i prawego boku"
80  INPUT XL, XR
90  IF XL >= XR OR XL < 0 OR XR > XM THEN 330
100 PRINT "Podaj wartości współrzędnych Y dla górnego i dolnego boku"
110 INPUT YT,YB
120 IF YT >= YB OR YT < 0 OR YB > YM THEN 330
130 DX = 1
140 DY = 1
150 X = XL + INT((XR - XL) / 2)
160 Y = YT + INT((YB - YT) / 2)     'Punkt początkowy w środku prostokąta
170 CLEARSCREEN
180 GRAPHICS
190      '********************* Rysuj prostokąt ***********************
200 DRAWLINE XL,YT TO XR,YT
210 DRAWLINE XR,YT TO XR,YB
220 DRAWLINE XR,YB TO XL,YB
230 DRAWLINE XL,YB TO XL,YT
240      '********************* Odbijaj punkt *************************
250 POINTPLOT X,Y
260 IF X = XL+1 OR X = XR-1 THEN DX = -DX     'Jeżeli punkt jest oddalony
270 IF Y = YT+1 OR Y = YB-1 THEN DY = -DY     'o jeden piksel od boku,
280 POINTOFF X,Y     'Usuń bieżący punkt                'zmień kierunek
290 X = X + DX       'Oblicz nowy punkt
300 Y = Y + DY
310 GOTO 250
320      '***********************************************************
330 PRINT "Błąd w wyborze boków prostokąta"
340 END
```

Wybierając większe wartości DX i DY możemy przyspieszyć proces odbijania w programie 7.2. Pięciokrotne zwiększenie wartości przyrostów powoduje pięciokrotny wzrost prędkości ruchu. Jeżeli jeden przyrost będzie większy niż drugi (np. DX > DY), to punkt będzie się poruszał szybciej w kierunku, dla którego przyrost jest większy. Alternatywnie, możemy zwolnić ruch w jednym kierunku ustalając wartość przyrostu dla tego kierunku mniejszą niż jeden (np. 0,5). Zmiana wartości przyrostu w czasie wykonywania programu może przyspieszać i zwalniać prędkość odbijania piksela. Jeżeli w czasie odbicia od ściany zmienia się wartość tylko jednego przyrostu, otrzymujemy efekt poślizgu.

Program 7.2 można tak modyfikować, aby działał dla dowolnych wartości przyrostów DX lub DY. Wymaga to zmiany warunku odbicia. Kierunek ruchu piksela trzeba zmienić wtedy, kiedy albo współrzędna X, albo współrzędna Y przekracza brzeg prostokąta. Możemy dokonać tej zmiany w programie 7.2, zastępując wiersze 260 i 270 następującymi:

```
260 IF DX > 0 THEN 268       'Ruch w prawo
262         'w przeciwnym przypadku ruch w lewo
263         'i DX jest ujemne
264 IF X + DX <= XL THEN DX = -DX
266 GOTO 269
268 IF X + DX >= XR THEN DX = -DX
269         'Test dla Y
270 IF DY > 0 THEN 278       'Ruch w dół
272         'w przeciwnym przypadku ruch w górę
273         'i DY jest ujemne
274 IF Y + DY <= YT THEN DY = -DY
276 GOTO 280
278 IF Y + DY >= YB THEN DY = -DY
```

Podany fragment programu zapewnia odbicie piksela wcześniej, nim osiągnie on bok prostokąta, jeżeli albo DX, albo DY jest większe niż 1. W celu uzyskania bardziej realistycznego odbicia możemy przed zmianą kierunku ruchu rysować piksel blisko brzegu. W tym celu należy tak wybierać wartości przyrostów i wymiary prostokąta, aby długość każdego boku była wielokrotnością przyrostu w danym kierunku. Jest to prawdopodobnie trochę zbyt duże ograniczenie dla wielu zastosowań. Bardziej ogólne rozwiązanie tego problemu polega na przedłużaniu toru ruchu piksela do przecięcia z bokiem i wymazaniu punktu przecięcia. Możemy wtedy uzyskać efekt odbijania piksela od boku dla dowolnie wybranego przyrostu.

Na rysunku 7.2 pokazano drogę piksela w kierunku boku pionowego. Startując z punktu początkowego (X, Y), punkt przecięcia (XI, YI) z brzegiem określa się z równania linii prostej.

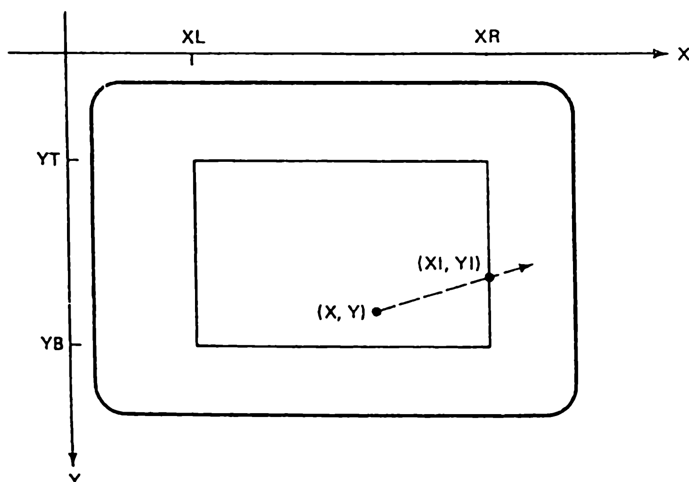
$$YI = M * XI + (Y - M * X) \tag{7.1}$$

przy czym XI = XR dla prawego boku i XI = XL dla lewego boku. Nachylenie M linii obliczamy korzystając z wartości przyrostów

$$M = DY/DX \tag{7.2}$$

Równania (7.1) i (7.2) stosuje się do określenia miejsca przecięcia dla dowolnego brzegu i dowolnego spośród możliwych kierunków ruchu. Dla przecięcia z górą lub dołem prostokąta XI oblicza się z (7.1) jako $XI = X + (YI - Y)/M$, przy czym $YI = YT$ albo $YI = YB$ (rys. 7.2). W każdym z tych punktów odbicia rysujemy piksel w odległości jednostkowej od brzegu i zmieniamy kierunek ruchu. W programie 7.3 skorzystano z tej metody do odbijania piłki.

W programie 7.3 przedstawiono piłkę bardziej realistycznie rysując ją jako koło wypełnione wybranym kolorem. Program można również tak zmodyfikować, aby zachować boki prostokąta, wymiar piłki i jej początkową pozycję.



Rys. 7.2. Piksel w pozycji (X, Y) wędrując wzdłuż linii ukośnej wewnątrz prostokąta przetnie prawy bok w punkcie (XI, YI)

Program 7.3. Odbijanie piłki wewnątrz prostokąta

```
10  'PROGRAM 7.3. Odbijanie piłki wewnątrz prostokąta
20       'DX = DY = 5. Jeżeli piłka zaraz wyjdzie poza granice prostoką-
30       'ta,to trzeba znaleźć przecięcie piłki z granicą,a potem nary-
40       'sować piłkę. Po zmianie kierunku kontynuować ruch piłki.
50       '*********************************************************
60  CLEARSCREEN
70  PRINT "Podaj wartości maksymalne poziomą i pionową"
80  PRINT "dla przyjętej rozdzielczości"
90  INPUT XM, YM
100 PRINT "Podaj wartości współrzędnych X dla lewego i prawego boku"
110 INPUT XL, XR
120 IF XL >= XR OR XL < 0 OR XR > XM THEN 940
130 PRINT "Podaj wartości współrzędnych Y dla górnego i dolnego boku"
140 INPUT YT,YB
150 IF YT >= YB OR YT < 0 OR YB > YM THEN 940
```

Program 7.3 (cd.)

```
160 DX = 5              'Piłka przesuwa się o 5 pikseli w każdym kroku
170 DY = 5
180 R = 3               'R - promień piłki
190 XN = XL + INT((XR - XL) / 2)   'Srodek piłki w środku prostokąta
200 YN = YT + INT((YB - YT) / 2)
210 CLEARSCREEN
220 GRAPHICS
230      '******************** Rysuj prostokąt ***********************
240 DRAWLINE XL,YT TO XR,YT
250 DRAWLINE XR,YT TO XR,YB
260 DRAWLINE XR,YB TO XL,YB
270 DRAWLINE XL,YB TO XL,YT
280      '******************** Odbijaj piłkę ************************
290 COLOR 0,0
300 CIRCLEPLOT X,Y,R          'Usuń piłkę z bieżącej pozycji
310 COLOR 1,0
320 CIRCLEPLOT XN,YN R        'Rysuj piłkę w nowej pozycji
330 X = XN                    'Pamiętaj bieżącą pozycję w X i Y
340 Y = YN
350 BX = 0                    'BX,BY - wskaźniki określające bok,z którym
360 BY = 0                    'zetknie się piłka
370 M = DY / DX               'M - nachylenie toru piłki
380      '
390      '*********************************************************
400      'Czy piłka uderzy w bok pionowy?
410 IF DX > 0 THEN 480       'Piłka przesuwa się w prawo
420      'W przeciwnym przypadku piłka przesuwa się w lewo (DX < 0)
430 IF X + DX -R > XL THEN 530     'Wciąż wewnątrz prostokąta
440 BX = 1                    'Wyjście z prostokąta przez bok X
450 XN = XL + R + 1           'Nowa współrzędna X jest przy lewym boku
460 GOTO 530
470      '^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
480      'Ruch w prawo
490 IF X + DX + R < XR THEN 530  'Wciąż wewnątrz prostokąta
500 BX = 1                    'Wyjście z prostokąta przez bok X
510 XN = XR - R - 1           'Nowa współrzędna X jest przy prawym boku
520      '*********************************************************
530      'Czy piłka uderzy w poziomy bok?
540 IF DY > 0 THEN 610       'Piłka porusza się do dołu
550      'W przeciwnym przypadku piłka porusza się do góry
560 IF Y + DY -R > YT THEN 670   'Wciąż wewnątrz prostokąta
570 BY = 1                    'Wyjście z prostokąta przez bok Y
580 YN = YT + R + 1           'Nowa współrzędna Y jest przy górnym boku
590 GOTO 670
600      '^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
610      'Ruch w dół
620 IF Y + DY + R < YB THEN 670  'Wciąż wewnątrz prostokąta
630 BY = 1                    'Wyjście z prostokąta przez bok Y
640 YN = YB - R - 1           'Nowa współrzędna Y jest przy dolnym boku
650      '
660      '*********************************************************
670      'Czy piłka odbije się od boku X, boku Y, obu boków, czy wcale?
680 IF BX = 0 AND BY = 0 THEN 730       'Nie ma odbicia
690 IF BX = 0 AND BY = 1 THEN 770       'Odbicie od Y
700 IF BX = 1 AND BY = 0 THEN 810       'Odbicie od X
710 IF BX = 1 AND BY = 1 THEN 850       'Odbicie od X i Y (narożnik)
720 '
730 '********************** Nie ma odbicia ***************************
740 XN = X + DX
750 YN = Y + DY
760 GOTO 290
770 '*********************. Odbicie od boku Y ************************
780 XN = (YN - Y) / M + X
790 DY = -DY
```

Program 7.3 (cd.)

```
800 GOTO 290
810 '********************* Odbicie od boku X ***************************
820 YN = (XN - X) * M + Y
830 DX = -DX
840 GOTO 290
850 '********************* Wejście w narożnik **************************
860       'Który bok będzie uderzony wcześniej?
870 IF ABS(XN - X) < ABS(YN - Y) THEN 810        'Odbicie od X
880 IF ABS(YN - Y) < ABS(XN - X) THEN 770        'Odbicie od Y
890       'Piłka jest w jednakowej odległości od boków X i Y narożnika
900 DX = -DX
910 DY = -DY
920 GOTO 290
930 '*******************************************************************
940 PRINT "Błąd w wyborze boków prostokąta"
950 END
```

Ruch wzdłuż linii krzywych

Ruch punktu lub okręgu wzdłuż pewnego toru zakrzywionego można uzyskać wyznaczając kolejne pozycje z równania tego toru. Ruch rozpoczyna się w zadanym punkcie początkowym i kończy w zadanym punkcie końcowym. Punkt możemy poruszać wzdłuż okręgu lub elipsy korzystając z metod rysowania okręgu omówionych w p. 5.1 albo z metod obracania przedstawionych w p. 6.3. W celu uzyskania ruchu jednostajnego po okręgu powinniśmy rozmieścić piksele równomiernie wzdłuż toru. Oznacza to, że powinniśmy do obliczania współrzędnych korzystać z równań trygonometrycznych i przyjmować jednakowe odstępy kątowe między kolejnymi pozycjami. W celu uzyskania szybszego ruchu wybiera się większe przyrosty kątowe (np. 30°). Zmieniając długość promienia okręgu otrzymamy ruch po spirali. Ruch obrotowy stosuje się przy modelowaniu orbit satelitów, systemu słonecznego lub części maszyn.

Piksel poruszający się wzdłuż paraboli symuluje ruch obiektu w rzucie ukośnym. Zasięg rzutu zależy od prędkości początkowej i kąta (względem poziomu), pod którym obiekt wyrzucono (rys. 7.3). Znając prędkość początkową S i kąt A można obliczyć zasięg R i największą wysokość HT toru jako

$$\begin{aligned} R &= S * S * SIN(2 * A)/G \\ HT &= ((S * SIN(A))\hat{}\ 2)/2 * G \end{aligned} \qquad (7.3)$$

przy czym G jest przyspieszeniem ziemskim (980 cm/s²). Przy ustalonej wartości S największy zasięg otrzymuje się dla kąta 45°. Wysokość HT jest największa, jeżeli wyrzucimy obiekt pionowo w górę (A = 90°). Pozycje piksela wzdłuż krzywej otrzymujemy zmieniając X od 0 do R i obliczając odpowiednie wartości Y z równania

$$Y = C1 * X\hat{}\ 2 + C2 * X + Y0 \qquad (7.4)$$

w którym Y0 jest wybranym przez nas punktem ekranu, a stałe C1 i C2 są określo-

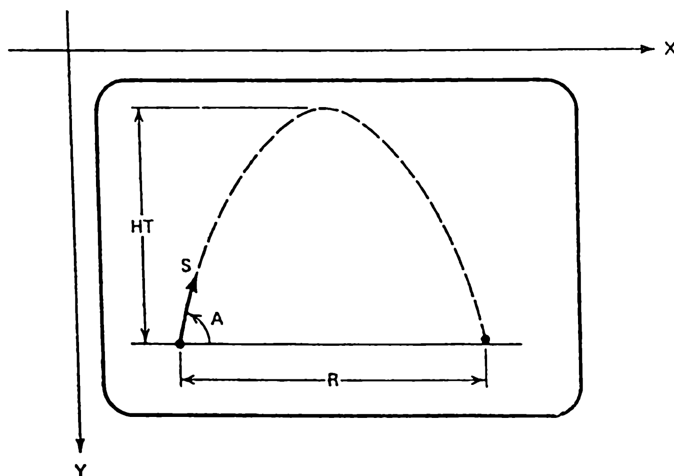
ne jako:

$$C1 = G/(2 * (S * COS(A))\hat{\ } 2)$$
$$C2 = -TAN(A) \tag{7.5}$$

Ruch punktu wzdłuż paraboli można uzyskać za pomocą programu 7.4 W danych wejściowych określa się punkt początkowy (X0, Y0), kąt rzutu A oraz prędkość początkową S. Wartość kąta A powinna być przyjmowana w zakresie od 0 do 90°, co oznacza, że ruch odbywa się w prawo (rys. 7.3). Dla wartości S z przedziału 200 do 600 otrzymuje się krzywe o wartościach wysokości HT oraz zasięgu R wynoszących ok. 200. Inne wartości zakresu przy ustalonej prędkości początkowej S uzyskuje się zmieniając wartość przyspieszenia G. W programie możemy zmieniać punkty początkowy i końcowy, co umożliwia symulowanie innych ruchów podobnego typu. Wyrzucenie obiektu z dachu budynku lub ze wzgórza oznacza, że ruch rozpoczyna się i kończy przy różnych wartościach Y. Przy symulacji toru obiektu spadającego z lecącego samolotu obiekt zaczyna ruch na najwyższej wysokości HT i spada na ziemię.

Program 7.4. Ruch punktu wzdłuż paraboli

```
10  'PROGRAM 7.4. Ruch wzdłuż paraboli
20  CLEARSCREEN
30  PRINT "Podaj wartości maksymalnę poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM, YM
60  PRINT "Podaj współrzędne punktu początkowego"
70  INPUT XO, YO
80  IF XO >= 0 AND XO <= XM AND YO >= 0 AND YO <= YM THEN 110
90  PRINT "Podaj ponownie współrzędne punktu  początkowego"
100 GOTO 60
110 PRINT "Podaj kąt rzutu (0 - 90)
120 INPUT A
130 A = A * 3.14159 / 180        'Zamiana wartości kąta A na radiany
140 PRINT "Podaj prędkość (100 - 600)"
150 INPUT S
160 G = 980                      'G - przyspieszenie ziemskie
170 R = S * S * SIN(2 * A) / G   'R - zasięg ruchu wzdłuż osi X
180 IF XO + R <= XM THEN 210     'Czy cała krzywa zmieści się na ekranie?
190 PRINT "Podaj ponownie kąt i prędkość"       'Jeżeli nie,wprowadź nowe
200 GOTO 110                                                   'wartości
210 HT = ((S * SIN(A)) ^ 2) / (2 * G)           'HT - wysokość krzywej
220 IF HT > 0 AND HT <= YM THEN 260        'Czy cała krzywa zmieści się na
230 PRINT "Podaj ponownie kąt i prędkość"                       'ekranie?
240 GOTO 110
250       'Oblicz współczynniki równania
260 C1 = G / (2 * (S * COS(A)) ^ 2)
270 C2 = - TAN(A)
280 CLEARSCREEN
290 GRAPHICS
300       '*************** Przesuwaj punkt wzdłuż krzywej **************
310 FOR X = 0 TO R
320   Y = C1 * X * X + C2 * X + YO
330   POINTPLOT X + XO,Y
340   POINTOFF X + XO,Y
350 NEXT X
360 END
```

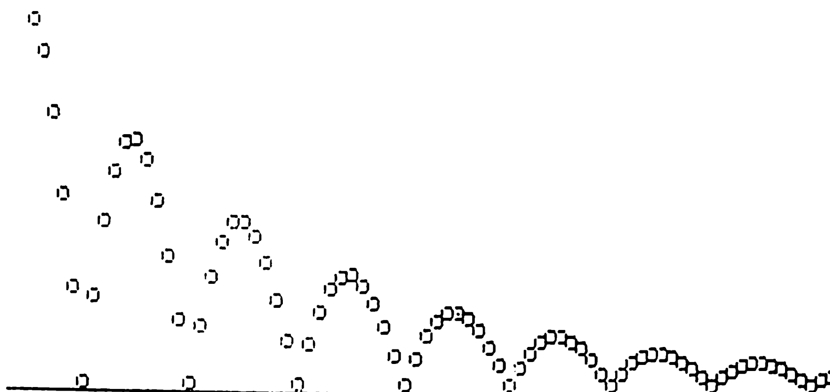


Rys. 7.3. W rzucie ukośnym z prędkością początkową S i kątem A obiekt osiągnie wysokość HT i spadnie w odległości R od punktu początkowego. Wysokość HT i zasięg R możemy obliczyć znając prędkość S oraz kąt A

Symulację odbijającej się piłki realizuje program 7.5. Piłka wyrzucana z wysokości H odbija się i przesuwa wzdłuż ekranu. Po każdym odbiciu zmniejsza się trochę wysokość wznoszenia się piłki (rys. 7.4). W celu uzyskania ruchu oscylacyjnego korzysta się z funkcji SIN, a w celu zmniejszania amplitudy z funkcji EXP.

Program 7.5. Symulacja odbić piłki wyrzuconej poziomo z pewnej wysokości

```
10  'PROGRAM 7.5. Odbijanie piłki spadającej z pewnej wysokości
20       'Program symuluje odbicia piłki.
30  CLEARSCREEN
40  PRINT "Podaj wartości maksymalne poziomą i pionową"
50  PRINT "dla przyjętej rozdzielczości"
60  INPUT XM, YM
70  PRINT "Z jakiej wysokości piłka jest wyrzucona?";
80  INPUT H
90  W = 3.14159 / 40           '40 - odległość między punktami odbicia
100 D = 90 * 3.14159 / 180     'Przesunięcie o 90 stopni (w radianach)
110 K = 0.01                   'K - współczynnnik tłumienia
120 CLEARSCEEN
130 GRAPHICS
140 DRAWLINE 0,YM TO XM,YM     'Rysuj podłoże
150 '
160      '******************** Rzuć piłkę i odbijaj ******************
170 FOR XN = 0 TO XM-10 STEP 4  'Wybrano krok 4,gdyż jest podzielni-
180   YN = H * SIN(W * XN + D) * EXP(-K * XN)              'kiem 40
190   YN = YM - ABS(YN) - 3
200   COLOR 0,0
210   CIRCLEPLOT X,Y,2         'Usuń bieżącą pozycję
220   COLOR 1,0
230   CIRCLEPLOT XN+10,YN,2    'Rysuj w nowym miejscu
240   X = XN + 10              'Zapamiętaj bieżącą pozycję w X i Y
250   Y = YN
260 NEXT XN
270 END
```



Rys. 7.4. Ruch piłki odcinającej się od lewej do prawej strony ekranu — obraz uzyskany za pomocą programu 7.5

Jeśli ruch piłki ma zaczynać się w punkcie o współrzędnych (0, H), przyjmujemy przesunięcie fazy w równaniu ruchu równe D = PI/2. Następnie wybieramy odległość (równą 40) między kolejnymi odbiciami tak, aby na ekranie można było zaobserwować kilka odbić. W tym celu przyjmujemy W = PI/40. Wreszcie, przy każdym odbiciu wyświetlamy piłkę dokładnie wtedy, kiedy uderza w podłoże. Oznacza to, że tak trzeba wybrać wartość przyrostu X, aby był on podzielnikiem 40. W tym przykładzie przyjmujemy X = 4 i piłka jest wyświetlana za każdym razem, gdy się odbija od podłoża dla X = 20, 60, 100 itd. Końcowy obraz jest przesunięty o 10 pikseli w prawo, aby ruch nie zaczynał się na krawędzi ekranu.

W animacji obrazów często stosuje się okręgi. Ruch kół można symulować przesuwając ich środki i obracając szprychy. Zmieniając wymiary okręgu możemy symulować ruch w naszą stronę lub w stronę przeciwną. Ruch wzdłuż dowolnego toru można uzyskać metodą interakcyjną. Korzystając z klawiatury, manipulatorów i pióra świetlnego możemy przesuwać obiekty w różnych kierunkach z różnymi prędkościami.

7.2. Odcinki i wielokąty

Prosty ruch odcinka lub wielokąta na ekranie uzyskuje się za pomocą takich samych podstawowych metod, jakie stosowano do poruszania punktu lub okręgu. Wyświetlamy wszystkie odcinki obiektu, wymazujemy obiekt i ponownie rysujemy wszystkie odcinki w nowym miejscu. Powtarzanie tej sekwencji stwarza wrażenie ruchu obiektu. Uwzględnienie przy animacji obiektu jego obrotów i skalowania albo dopuszczenie niezależnego ruchu jego części składowych wymaga dalszych rozważań.

Odcinki

Odcinek narysowany między punktami (X, Y1) i (X, Y2) można poruszać w kierunku pionowym zmieniając tylko wartość współrzędnej Y. Program 7.6 odbija pionowy odcinek w górę i w dół między dwoma ustalonymi punktami. Ruch takiego odcinka można przyspieszać lub opóźniać zmieniając wartość przyrostu DY w przedziale 1 do 20. Program ten nie dopuszcza do osiągnięcia lub przekroczenia granic górnej (YT) i dolnej (YB).

Program 7.6. Odbijanie odcinka w pionie

```
10  'PROGRAM 7.6. Przesuwanie pionowego odcinka w górę i w dół
20  CLEARSCREEN
30  PRINT "Podaj wartości największe poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM, YM
60  PRINT "Podaj wartości Y dla górnej i dolnej granicy odbić"
70  INPUT YT, YB
80  IF YT < YB AND YT >= 0 AND YB <= YM THEN 110
90  PRINT "Złe granice. Wybierz YT < YB - obie wartości z przedziału";
95  PRINT "0 do YM";
100 GOTO 60
110 PRINT "Podaj wartość przyrostu Y"
120 INPUT DY
130 X = INT(XM / 2)                'Umieść odcinek w środku ekranu
140 Y1 = YT + INT((YB - YT) / 2) - 20      'Końce odcinka są 20 pikseli
150 Y2 = YT + INT((YB - YT) / 2) + 20      'powyżej i poniżej środka
160 CLEARSCREEN                            'granic
170 GRAPHICS
180      '*********************************************************
190 COLOR 1,0
200 DRAWLINE X,Y1 TO X,Y2
210 IF DY > 0 THEN 250             'Ruch odcinka w dół
220      'W przeciwnym wypadku ruch do góry (DY jest ujemne)
230 IF Y1 + DY <= YT THEN DY = -DY         'Zmiana kierunku
240 GOTO 270
250      'Ruch odcinka w dół
260 IF Y2 + DY >= YB THEN DY = -DY         'Zmiana kierunku
270      'Usuń bieżący odcinek, oblicz nową pozycję, rysuj
280 COLOR 0,0
290 DRAWLINE X,Y1 TO X,Y2
300 Y1 = Y1 + DY
310 Y2 = Y2 + DY
320 GOTO 180
330 END
```

Ruch odcinka pionowego w kierunku poziomym możemy uzyskać zmieniając wartość współrzędnej X zamiast Y. W tym przypadku można realizować ruch odcinka między pionowymi ścianami, umieszczonymi w punktach o współrzędnych XL i XR. Zmieniając obie współrzędne uzyska się ruch ukośny odcinka. Można wtedy odbijać odcinek wewnątrz prostokąta, podobnie jak pojedynczy piksel. Odcinki poziome lub ukośne porusza się w podobny sposób. Przesuwamy oba końce odcinka o takie same przyrosty DX i DY. Dla większych wartości przyrostów odcinek porusza się szybciej. Zmianę kierunku uzyskuje się zawsze przy zmianie znaku jednego lub obu przyrostów.

Odcinki można poruszać wzdłuż krzywej zmieniając wartości DX i DY w czasie ruchu odcinka. Poruszając odcinek wzdłuż określonej krzywej utrzymujemy ustalony punkt odcinka na krzywej w ciągu całego ruchu. Na przykład, program 7.7 przesuwa jeden koniec odcinka poziomego wzdłuż okręgu. Odcinek porusza się tam i z powrotem, do góry i do dołu symulując ruch poziomego drążka po powierzchni bocznej walca (rys. 7.5).

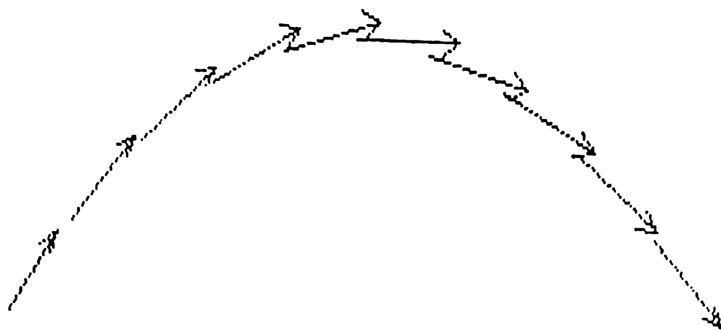
Program 7.7. Ruch odcinka wzdłuż okręgu

```
10  'PROGRAM 7.7. Przesuwanie odcinka wzdłuż okręgu
20  CLEARSCREEN
30  PRINT "Podaj wartości największe  poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM, YM
60  XC = INT(XM / 2) - 40      'XC,YC - środek okręgu
70  YC = INT(YM / 2)
80  R = 50                     'R - promień
90  YA = 5/6                   'YA - współczynnik korekcji rozdzielczości
100      'Obliczaj punkty X1 co 15 stopni wzdłuż okręgu
110 RE = 360 * 3.14159 / 180        'Zamiana kąta 360 stopni na radiany
120 DA =  15 * 3.14159 / 180        'Zamiana kąta  15 stopni na radiany
130 CLEARSCREEN
140 GRAPHICS
150      '****************************************************************
160 FOR A = DA TO RE STEP DA
170   COLOR 0,0
180   DRAWLINE X,Y TO X + 80,Y      'Usuń bieżący odcinek
190   X = XC + R * COS(A)
200   Y = YC + R * SIN(A) * YA
210   COLOR 1,0
220   DRAWLINE X,Y TO X + 80,Y      'Odcinek ma długość 80 jednostek
230 NEXT A
240 GOTO 160
250 END
```

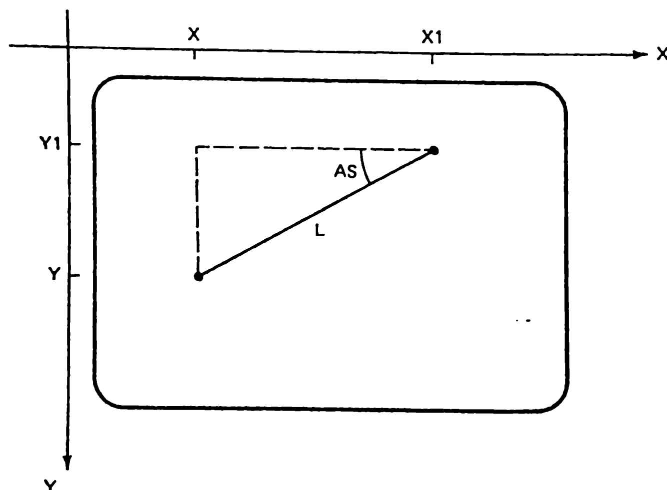


Rys. 7.5. Pozycje odcinka, którego lewy koniec porusza się po okręgu (program 7.7). Poziomy odcinek startuje z pozycji z prawej strony, porusza się w dół i w lewo, a następnie wzdłuż okręgu z powrotem do pozycji początkowej

Niekiedy zachodzi potrzeba, aby odcinek zmieniał kąt nachylenia w miarę przesuwania się wzdłuż krzywej. Na rysunku 7.6 przedstawiono kolejne pozycje i nachylenia odcinka (strzały z grotem), którego jeden koniec porusza się po paraboli i którego nachylenie zmienia się tak, aby w każdym kroku odcinek był styczny do krzywej. Taki ruch może symulować lot strzały w powietrzu. W celu uzyskania takiego ruchu musimy zmieniać nachylenie odcinka w czasie ruchu. Dla paraboli



Rys. 7.6. Ruch odcinka, którego koniec przesuwa się wzdłuż paraboli od lewej do prawej strony i który stale jest styczny do toru (program 7.8)



Rys. 7.7. Wartości współrzędnych (X1, Y1) jednego końca odcinka można obliczyć znając długość odcinka L, kąt nachylenia AS oraz współrzędne (X, Y) drugiego końca odcinka. Kąt nachylenia AS określa się znając nachylenie M ze wzoru AS = ATN(M)

o równaniu (7.4) nachylenie stycznej do krzywej w punkcie X wyznacza się z zależności

$$M = 2 * C1 * X + C2 \tag{7.6}$$

Współrzędne (X1, Y1) drugiego końca odcinka można określić znając nachylenie i długość odcinka (rys. 7.7). Z zależności wiążących boki trójkąta z przeciwprostokątną L otrzymujemy:

$$\begin{aligned} X1 - X &= L * COS(AS) \\ Y1 - Y &= L * SIN(AS) \end{aligned} \tag{7.7}$$

Kąt AS jest kątem nachylenia odcinka względem poziomu. Kąt ten obliczamy z zależności

$$AS = ATN(M) \tag{7.8}$$

Z równań (7.7) i (7.8) korzystano w programie 7.8 w celu uzyskania efektu ruchu strzały wzdłuż paraboli (rys. 7.6). Grot strzały zaczepiony w punkcie (X1, Y1) rysujemy jako dwa krótkie odcinki. Nachylenia tych krótkich odcinków niewiele różnią się od nachylenia M strzały. W programie jedno z tych nachyleń jest o 0,75 większe niż M, a drugie o 0,75 mniejsze niż M. Długość każdego małego odcinka wynosi 8, a długość strzały 40.

Za pomocą metody użytej w programie 7.8 otrzymuje się ruch odcinka stycznego do dowolnej krzywej. Jeden koniec odcinka przesuwamy wzdłuż krzywej.

Program 7.8. Ruch strzały wzdłuż paraboli

```
10  'PROGRAM 7.8. Ruch strzały wzdłuż paraboli
20       'Kolejno jest rysowana i wymazywana strzała, której koniec
30       'jest punktem paraboli. Pozostałą część strzały określa się
40       'znając współrzędne jej końca, nachylenie stycznej do toru w
50       'tym punkcie oraz długość strzały
60  '
70  '**************************************************************
80  CLEARSCREEN
90  PRINT "Podaj wartości największe  poziomą i pionową"
100 PRINT "dla przyjętej rozdzielczości"
110 INPUT XM, YM
120 PRINT "Podaj współrzędne punktu początkowego"
130 INPUT XO, YO
140 IF XO>=0 AND XO<=XM AND YO>=0 AND YM<=YM THEN 170
150 PRINT "Punkt początkowy poza ekranem"
160 GOTO 120
170 PRINT "Podaj kąt strzału (0 - 90)";
180 INPUT A
190 A = A * 3.14159 / 180       'Zamiana wartości kąta A na radiany
200 PRINT "Podaj prędkość (100 - 600)";
210 INPUT S
220 G = 980                     'G - przspieszenie ziemskie
230 LA = 40                     'LA - długość strzały
240 LT = 8                      'LT - długość grotu strzały
250      'Wyznacz zasięg i wysokość lotu strzały
260 R = S * S * SIN(2 * A) / G  'R - zasięg lotu strzały
270 IF XO + R <= XM THEN 300    'Czy koniec toru na ekranie?
280 PRINT "Podaj ponownie wartości kąta i prędkości"
290 GOTO 170
```

Program 7.8 (cd.)

```
300 HT = ((S * SIN(A)) ^ 2) / (2 * G)    'HT - wysokość toru strzały
310 IF HT > 0 AND HT <= YM THEN 340     'Czy góra toru na ekranie?
320 PRINT "Wprowadź ponownie wartości kąta i prędkości"
330 GOTO 170
340      'Określ współczynniki równania paraboli
350 C1 = G / (2 * (S * COS(A)) ^ 2)
360 C2 = -TAN(A)
370 CLEARSCREEN
380 GRAPHICS
390      '********************* Ruch strzały ************************
400      'Wyznaczaj wzdłuż paraboli koniec strzały i rysuj ją
410 FOR X = 0 TO R STEP R / 10          'Ustawiaj strzałę w kolejnych
420   Y = C1 * X * X + C2 * X + Y0      'pozycjach (co 0.1 R)
430   'X,Y - współrzędne końca strzały leżącego na paraboli
440   'Znajdź drugi koniec strzały
450   M = 2 * C1 * X + C2               'M - nachylenie strzały
460   A1 = ATN(M)                 'Kąt A1 oblicza się jako arctg(M)
470   Y1 = Y + LA * SIN(A1)
480   X1 = X + LA * COS(A1)
490   IF X1 > XM OR Y1 > YM THEN 800    'Czy drugi koniec na ekranie?
500   GOSUB 730                   'Usuń strzałę
510   'Obliczenia dla grotu strzały
520   M2 = M + 0.75               'Nachylenie jednego odcinka grotu
530   A2 = ATN(M2)
540   X2 = X1 - LT * COS(A2)
550   Y2 = Y1 - LT * SIN(A2)
560   M3 = M - 0.75               'Nachylenie drugiego odcinka grotu
570   A3 = ATN(M3)
580   X3 = X1 - LT * COS(A3)
590   Y3 = Y1 - LT * SIN(A3)
600   GOSUB 670                   'Rysuj strzałę
610   XS = X             'Zapamiętaj bieżącą pozycję w XS,YS,X1S,Y1S
620   YS = Y
630   X1S = X1
640   Y1S = Y1
650 NEXT X
660 GOTO 800
670 '*********************** Rysuj strzałę **************************
680 COLOR 1,0
690 DRAWLINE X,Y TO X1,Y1
700 DRAWLINE X1,Y1 TO X2,Y2
710 DRAWLINE X1,Y1 TO X3,Y3
720 RETURN
730 '*********************** Usuń strzałę ***************************
740 COLOR 0,0
750 DRAWLINE XS,YS TO X1S,Y1S
760 DRAWLINE X1S,Y1S TO X2,Y2
770 DRAWLINE X1S,Y1S TO X3,Y3
780 RETURN
790 '*************************************************************
800 END
```

Dla każdej pozycji obliczamy kąt nachylenia oraz współrzędne drugiego końca odcinka korzystając z równania (7.7). Nachylenie odcinka stycznego do krzywej w dowolnym punkcie określa się z równania tej krzywej. Na przykład, dla równania wielomianowego trzeciego stopnia o postaci

$$Y = C1 * X\hat{}3 + C2 * X\hat{}2 + C3 * X + C4 \tag{7.9}$$

nachylenie w punkcie X określa się na podstawie pochodnej tego równania

$$M = 3 * C1 * X\hat{}2 + 2 * C2 * X + C3 \tag{7.10}$$

Dla sinusoidy

$$Y = H * SIN(W * X + D) \tag{7.11}$$

nachylenie w punkcie X oblicza się jako

$$M = H * W * COS(W * X + D) \tag{7.12}$$

Ruch odcinka stycznego do okręgu lub elipsy można uzyskać korzystając z transformacji obrotu.

Ruch obrotowy odcinka wokół ustalonego punktu otrzymuje się stosując kolejno transformacje obrotu. Program 7.9 daje w wyniku obrót odcinka wokół jego środka. Taki rodzaj ruchu może być użyty do reprezentacji szprychy obracającego się koła. Równania obrotu stosowane w tym programie przekształcono tak, by skrócić czas obliczeń.

Program 7.9. Wirujący odcinek

```
10  'PROGRAM 7.9. Obracanie odcinka wokół jego środka
20  CLEARSCREEN
30  PRINT "Podaj wartości największe poziomą i pionową"
40  PRINT "dla przyjętej rozdzielczości"
50  INPUT XM, YM
60  XC = INT(XM / 2)        'Punkt centralny ekranu jest środkiem obrotu
70  YC = INT(YM / 2)
80  X1 = XC - 10            'Długość odcinka wynosi 20 jednostek
90  X2 = XC +10
100 Y1 = YC
110 Y2 = YC
120 YA = 5/6                'YA - współczynnik korekcji dla wartości Y
130 XA = 6/5                'XA - współczynnik korekcji dla wartości X
140 A = 15 * 3.14159 / 180  'Zamiana 15 stopni na radiany
150        'Oblicz stałe dla równania obrotu
160 CA = COS(A)
170 SX = SIN(A) * XA
180 SY = SIN(A) * YA
190 XE = XC - XC * CA - YC * SX    'Znajdź stałe elementy równań obrotu
200 YE = YC - YC * CA + XC * SY
210 CLEARSCREEN
220 GRAPHICS
230        '***************** Obracaj końce i rysuj *******************
240 X1R = XE + X1 * CA + Y1 * SX
250 Y1R = YE + Y1 * CA - X1 * SY
260 X2R = XE + X2 * CA + Y2 * SX
270 Y2R = YE + Y2 * CA - X2 * SY
280        'Usuń stary odcinek
290 COLOR 0,0
300 DRAWLINE X1,Y1 TO X2,Y2
310        'Rysuj nowy odcinek
320 COLOR 1,0
330 DRAWLINE X1R,Y1R TO X2R,Y2R
340        'Zapamiętaj bieżące punkty do późniejszego wymazania
350 X1 = X1R
360 Y1 = Y1R
370 X2 = X2R
380 Y2 = Y2R
390 GOTO 230
400 END
```

Poruszające się odcinki mogą być w różny sposób wykorzystane do animacji obrazów. W grach ruchome odcinki mogą reprezentować packę lub rakietę i uderzać odbijaną piłkę. Możemy również w różny sposób poruszać poszczególne odcinki składowe obrazu, aby móc reprezentować bardziej złożony ruch, tak jak na przykład przy symulacji ruchu postaci.

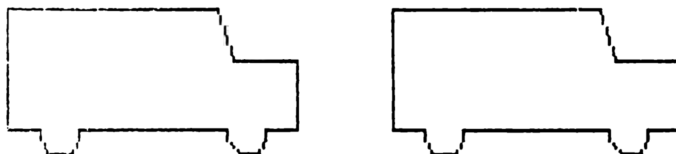
Wielokąty

Wielokąt jest określany przez współrzędne wierzchołków, które powinny być pamiętane w takiej kolejności, w jakiej mają być łączone. Przesuwanie wielokąta oznacza przesuwanie wszystkich punktów wierzchołkowych i ponowne rysowanie wielokąta. Program 7.10 przesuwa sylwetkę ciężarówki z lewej na prawą stronę

Program 7.10. Ruch ciężarówki wzdłuż prostej

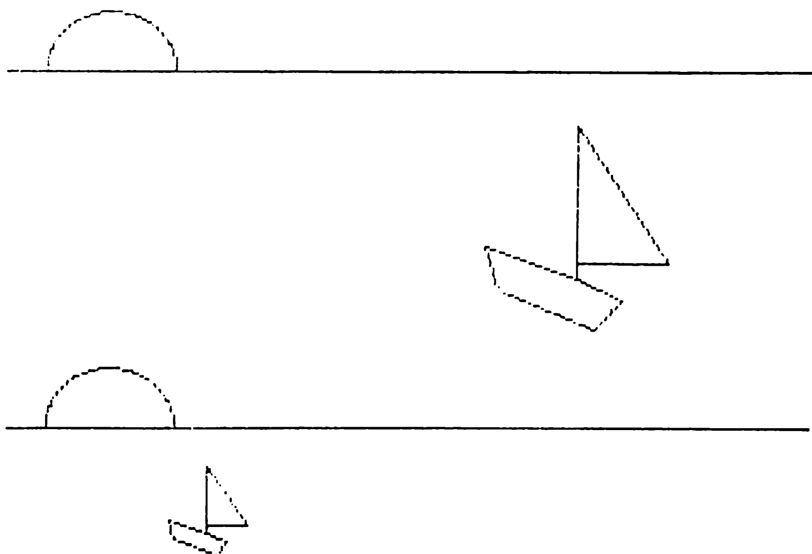
```
10  'PROGRAM 7.10. Ciężarówka poruszająca się w poprzek ekranu
20  CLEARSCREEN
30  DIM X(20), Y(20), XT(20)
40  XM = 300                    'XM - maksymalna wartość X w systemie
50  READ N                      'N - liczba punktów w obrazie
60  FOR K = 1 TO N
70    READ X(K), Y(K)
80    'Zapamiętaj X w innej tablicy - będzie potrzebne przy rysowaniu
90    XT(K) = X(K)
100 NEXT K
110 H = 15
120 GRAPHICS
130 GOSUB 260                   'Rysuj
140   '****************** Przesuń,usuń,rysuj ********************
150 FOR K = 1 TO N
160   XT(K) = X(K) + H          'Nie ma potrzeby przesuwać Y
170 NEXT K
180 GOSUB 320                   'Usuń obraz ze starej pozycji
190 GOSUB 260                   'Rysuj obraz w nowej pozycji
200 IF XT(14) + H > XM THEN 430   'Stop - jeżeli przesunięcie skrajnego
210       'prawego punktu powoduje wyjście poza ekran.Kontynuacja - w
220 FOR K = 1 TO N                'przeciwnym przypadku
230   X(K) = XT(K)              'Zapamiętaj bieżącą pozycję w X
240 NEXT K
250 GOTO 140
260 '*********************** Program rysowania ***********************
270 COLOR 1,0
280 FOR K = 1 TO N - 1
290   DRAWLINE XT(K),Y(K) TO XT(K+1),Y(K+1)
300 NEXT K
310 RETURN
320 '*********************** Program usuwania ************************
330 COLOR 0,0
340 FOR K = 1 TO N - 1
350   DRAWLINE X(K);Y(K) TO X(K+1),Y(K+1)
360 NEXT K
370 RETURN
380 '****************************************************************
390 DATA 19
400 DATA 10,90,20,90,20,93,22,97,30,97,32,93,32,90
410 DATA 78,90,78,93,82,97,88,97,90,93,90,90,100,90
420 DATA 100,70,80,70,75,55,10,55,10,90
430 END
```

ekranu, tak jak to zilustrowano na rys. 7.8. Program będzie przesuwał dowolny obiekt zdefiniowany w instrukcjach DATA; ale im więcej jest odcinków w obiekcie, tym wolniejsza jest animacja. Możemy uogólnić ten ruch w dowolnym kierunku wprowadzając, tak jak w poprzednich przykładach, wartości przyrostów w obu kierunkach poziomym i pionowym.



Rys. 7.8. Animacja ruchu ciężarówki w poprzek ekranu (program 7.10)

Skalowanie wielokąta symuluje ruch w stronę operatora lub w stronę przeciwną. Na rysunku 7.9 łódka żegluje w stronę słońca (program 7.11). Łódź jest sukcesywnie skalowana względem punktu na horyzoncie, co powoduje efekt jej oddalania się.



Rys. 7.9. Łódka żeglująca w stronę słońca. Animacja za pomocą programu 7.11, w którym wykorzystano transformacje skalowania

Program 7.11. Animacja przez skalowanie (żaglówka)

```
10  'PROGRAM 7.11. Żeglowanie w stronę słońca
20        'Łódź jest kolejno skalowana w odniesieniu do stałego punktu
30        'o współrzędnych (-30,70). Skalowanie zmniejsza łódź i przesu-
40        'wa ją od prawej do lewej strony.
50  CLEARSCREEN
60  DIM X(10), Y(10), XN(10), YN(10)
70  YA = 5/6          'YA - współczynnik korekcji rozdzielczości
80  GRAPHICS
90        '********************** Twórz tło **************************
100 DRAWLINE 0,60 TO 319,60      'Rysuj horyzont
110       'Rysuj słońce w postaci półokręgu
120 R = 25            'R - promień słońca
130 XS = 40           'XS,YS - środek słońca
140 YS = 59
150 DA = 1 / R        'DA - przyrost kąta
160 C1 = 180 * 3.14159 / 180
170 C2 = 180 * 3.14159 / 180
180 FOR A = C1 TO C2 STEP DA
190   POINTPLOT XS + R * COS(A), YS + R * SIN(A) * YA
200 NEXT A
210       '************** Odczytaj punkty łodzi i rysuj **************
220 FOR K = 1 TO 9
230   READ X(K), Y(K)
240   Y(K) = Y(K) * YA
250   XN(K) = X(K)
260   YN(K) = Y(K)
270 NEXT K
280 GOSUB 510         'Rysuj łódź
290       '****************** Płyń w stronę słońca *******************
300 XF = -30          'Skaluj w odniesieniu do XF,YF
310 YF = 70
320 HS = 0.9          'Zmniejsz łódź do 9/10 bieżących wymiarów
330 VS = 0.9
340       'Oblicz stałe elementy równań skalowania
350 XE = XF * (1 - HS)
360 YE = YF * (1 - VS)
370 FOR T = 1 TO 12
380   FOR K = 1 TO 9
390     XN(K) = X(K) * HS + XE + H
400     YN(K) = Y(K) * VS + YE + V
410   NEXT K
420   GOSUB 570       'Usuń starą pozycję
430   GOSUB 510       'Rysuj nową pozycję
440   FOR K = 1 TO 9           'Zapamiętaj bieżącą pozycję w X i Y
450     X(K) = XN(K)
460     Y(K) = YN(K)
470   NEXT K
480 NEXT T
490 GOTO 650
500 '
510 '************************* Rysuj łódź ****************************
520 COLOR 1,0
530 FOR K = 1 TO 8
540   DRAWLINE XN(K),YN(K) TO XN(K+1),YN(K+1)
550 NEXT K
560 RETURN
570 '************************* Usuń łódź *****************************
580 COLOR 0,0
590 FOR K = 1 TO 8
600   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
610 NEXT K
620 RETURN
630 '****************************************************************
640 DATA 250,148,290,148,250,80,250,156,270,167
650 DATA 258,182,215,162,210,140,250,156
660 END
```

7.3. Ruch złożony

Animację obrazów uzyskuje się poruszając równocześnie różne części obrazu. Przykładowe zastosowania to: pojazdy poruszające się w różnych kierunkach, zwierzęta z poruszającymi się kończynami, ludzie poruszający rękami i nogami, sieci elektryczne lub inne, w których symuluje się „przepływ" równocześnie w różnych gałęziach sieci, złożone urządzenie z wieloma poruszającymi się częściami. Przy poruszaniu różnych części obrazu ważne jest koordynowanie ruchów ze względu na zapewnienie jak największego realizmu modelu.

Program 7.12 tworzy platformę pokazaną na rys. 7.10. Koła platformy obracają się w miarę przesuwania platformy w lewo. Szprychy w każdym z kół są wymazywane tuż przed narysowaniem ich w nowej obróconej pozycji, tak aby czas wyświetlania pustych powierzchni był jak najkrótszy. W celu skrócenia czasu obliczeń ograniczamy liczbę szprych. Im więcej odcinków zawiera rysunek, tym dłużej trwa animacja.

Program 7.12. Ruch złożony: poruszająca się platforma z obracającymi się kołami

```
10   'PROGRAM 7.12. Ruch platformy i kół
20        'Program rysuje platformę i porusza ją od prawej do lewej
30        'strony. Szprychy kół obracają się dwa razy w czasie każdego
40        'ruchu platformy.
50   CLEARSCREEN
60   DIM X(20), Y(20), XN(20), YN(20)
70   YA = 5/6                    'YA - współczynnik korekcji dla wartości Y
80   XA = 6/5                    'XA - współczynnik korekcji dla wartości X
90   FOR K = 1 TO 18             'Odczytaj punkty platformy
100    READ X(K), Y(K)
110    Y(K) = Y(K) * YA
120    XN(K) = X(K)
130    YN(K) = Y(K)
140  NEXT K
150  H = -5                      'H - wartość przesunięcia w poziomie
160  A = 25 * 3.14159 / 180      'Zamień stopnie na radiany
170       'Oblicz stałe elementy równań rotacji
180  CA = COS(A)
190  SX = SIN(A) * XA
200  SY = SIN(A) * YA
210  GRAPHICS
220  GOSUB 360                   'Rysuj platformę
230       '***************** Przesuń, usuń i rysuj *********************
240  GOSUB 480                   'Obracaj szprychy
250  IF XN(2) + H < 0 THEN 940    'Jeżeli przesunięcie lewych skrajnych
260  FOR K = 1 TO 18              'punktów powoduje wyjście poza ekran, to
270    X(K) = XN(K)               'stop; w przeciwnym razie,kontynuuj
280    Y(K) = YN(K)
290  NEXT K
300  FOR K = 1 TO 18
310    XN(K) = X(K) + H          'Dodaj wielkość przesunięcia
320  NEXT K
330  GOSUB 770                   'Usuń bieżącą pozycję
340  GOSUB 360                   'Rysuj nową pozycję
350  GOTO 240
360  '*********************** Program rysowania ***********************
370  COLOR 1,0
380  FOR K = 1 TO 5
390    DRAWLINE XN(K),YN(K) TO XN(K+1),YN(K+1)
400  NEXT K
```

Program 7.12 (cd.)

```
410 DRAWLINE XN(7),YN(7) TO XN(8),YN(8)
420 CIRCLEPLOT XN(9),YN(9),11   'Srodki kół są elementami 9,10 w tablicy
430 CIRCLEPLOT XN(10),YN(10),11
440 FOR K = 11 TO 17 STEP 2     'Rysuj szprychy
450   DRAWLINE XN(K),YN(K) TO XN(K+1),YN(K+1)
460 NEXT K
470 RETURN
480 '******************** Program obrotu szprych ********************
490 YO = Y9                      'Współrzędna Y obu kół
500 FOR S = 1 TO 2               'Obróć szprychy dwa razy
510   FOR K = 11 TO 18           'Zapamiętaj bieżącą pozycję szprych w X,Y
520     X(K) = XN(K)
530     Y(K) = YN(K)
540   NEXT K
550   XO = XN(9)                 'Współrzędna X środka pierwszego koła
560   FOR K = 11 TO 14
570     XS = XN(K)               'Zapamiętaj XN(K) do obliczenia YN(K)
580     XN(K) = XO + (XN(K) - XO) * CA + (YN(K) - YO) * SX
590     YN(K) = YO + (YN(K) - YO) * CA + (XS - XO) * SY
600   NEXT K
610   XO = XN(10)                'Współrzędna X środka drugiego koła
620   FOR K = 15 TO 18
630     XS = XN(K)               'Zapamiętaj XN(K) do obliczenia YN(K)
640     XN(K) = XO + (XN(K) - XO) * CA + (YN(K) - YO) * SX
650     YN(K) = YO + (YN(K) - YO) * CA - (XS - XO) * SY
660   NEXT K
670   COLOR 0,0
680   FOR K = 11 TO 17 STEP 2          'Usuń bieżącą pozycję
690     DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
700   NEXT K
710   COLOR 1,0
720   FOR K = 11 TO 17 STEP 2          'Rysuj nową pozycję
730     DRAWLINE XN(K),YN(K) TO XN(K+1),YN(K+1)
740   NEXT K
750 NEXT S
760 RETURN
770 '********************** Program usuwania ***********************
780 COLOR 0,0
790 FOR K = 1 TO 5
800   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
810 NEXT K
820 DRAWLINE X(7),Y(7) TO X(8),Y(8)
830 CIRCLEPLOT X(9),Y(9),11
840 CIRCLEPLOT X(10),Y(10),11
850 FOR K = 11 TO 17 STEP 2
860   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
870 NEXT K
880 RETURN
890 '*****************************************************************
900 DATA 190,140,180,140,180,120,270,120,270,140,260,140,240,140,210,140
910 DATA 200,140,250,140
920 DATA 210,140,190,140,200,130,200,150
930 DATA 260,140,240,140,250,130,250,150
940 END
```



Rys. 7.10. Ruch złożony – animacja platformy za pomocą przesunięcia i obracania kół (program 7.12)

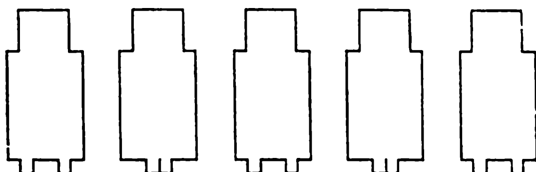
Program 7.13 rysuje kroczącego robota. Przykład ten łączy przesunięcie z poruszaniem stóp. Wynik działania programu dla pięciu pozycji ruchu robota sunącego w prawo zilustrowano na rys. 7.11.

Program 7.13. Ruch złożony: kroczący robot

```
10  'PROGRAM 7.13. Kroczący robot
20       'Robot porusza się od lewej do prawej strony ekranu. W każdym
30       'kroku nogi są przesuwane o wielkość zależną od wyświetlanej
40       'pozycji robota. W pozycji pierwszej nogi są złączone. W pozy-
50       'cji drugiej nogi są oddalone od siebie. Tułów robota jest
60       ,przesuwany o stałą odległość.
70
80  '*************************************************************************
90  CLEARSCREEN
100 DIM X(16), Y(16), XN(16), YN(16)
110 XM = 279                      'XM - maksymalna wartość X dla systemu
120 FOR K = 1 TO 16
130   READ XN(K),YN(K)
140 NEXT K
150 B = 15              'B - wielkość przesunięcia tułowia
160 P = 2               'P - wskazuje następną wyświetlaną pozycję
170 GRAPHICS
180 GOSUB 510                     'Rysuj
190       '************************* Przesuń ***************************
200       'Zapamiętaj bieżące punkty do późniejszego usunięcia
210 FOR K = 1 TO 16
220   X(K) = XN(K)
230   Y(K) = YN(K)
240 NEXT K
250 FOR K = 1 TO 8                'Przesuń punkty tułowia
260   XN(K) = X(K) + B
270 NEXT K
280 IF P = 2 THEN 340             'Czy robot przechodzi do pozycji drugiej?
290       'Jeśli nie przechodzimy do pozycji pierwszej
300 FI = 10             'FI - przyrost dla prawej stopy
310 RI = 20             'RI - przyrost dla lewej stopy
320 P = 2               'Następnym razem przejdź do pozycji drugiej
330 GOTO 380
340       'Przejście do pozycji drugiej
350 FI = 20
360 RI = 10
370 P = 1               'Następnym razem przejdź do pozycji pierwszej
380       'Przesuń lewą stopę
390 FOR K = 9 TO 12
400   XN(K) = X(K) + RI
410 NEXT K
420       'Przesuń prawą stopę
430 FOR K = 13 TO 16
440   XN(K) = X(K) + FI
450 NEXT K
460 GOSUB 580           'Usuń bieżącą pozycję
470 GOSUB 510           'Rysuj nową pozycję
480 IF XN(1) + B > XM THEN 680    'Jeżeli przesunięcie skrajnego prawego
490 GOTO 190                      'punktu powoduje wyjście poza ekran, to
500 '                             'stop; w przeciwnym razie kontynuuj
510 '************************ Program rysowania ************************
520 COLOR 1,0
530 FOR K = 1 TO 15
540   DRAWLINE XN(K),YN(K) TO XN(K+1),YN(K+1)
550 NEXT K
560 DRAWLINE XN(16),YN(16) TO XN(1),YN(1)
570 RETURN
```

Program 7.13 (cd.)

```
580 '************************ Program usuwania ***********************
590 COLOR 1,0
600 FOR K = 1 TO 15
610   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
620 NEXT K
630 DRAWLINE X(16),Y(16) TO X(1),Y(1)
640 RETURN
650 '*******************************************************************
660 DATA 40,150,40,110,35,110,35,95,15,95,15,110,10,110,10,150
670 DATA 20,150,20,155,25,155,25,150,25,150,25,155,30,155,30,150
680 END
```



Rys. 7.11. Ruch złożony – robot kroczący w poprzek ekranu; wynik działania programu 7.13

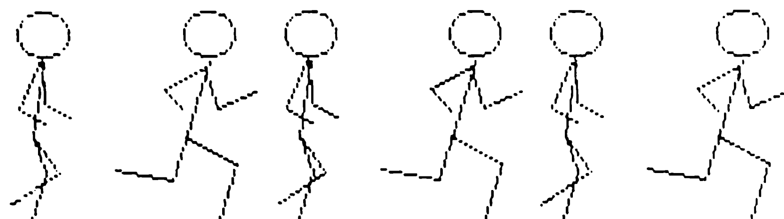
Program 7.14 rysuje biegnącą sylwetkę. Każda „ramka" była narysowana na papierze milimetrowym, a następnie przeniesiona do układu współrzędnych ekranu tak, aby stworzyć wrażenie ruchu rąk i nóg w biegu. Każda z dwóch pozycji biegacza jest pamiętana w oddzielnej tablicy. Umożliwia to przemienne przesuwanie każdej ramki wzdłuż ekranu bez konieczności przeliczania kolejnych pozycji części obrazu. Otrzymany ruch zilustrowano na rys. 7.12.

Program 7.14. Ruch złożony: sylwetka biegacza; wykorzystuje się dwie „ramki"

```
10  'PROGRAM 7.14. Biegacz
20      'Program rysuje biegacza przełączając dwie pozycje
30      '(lub ramki) przy ruchu wzdłuż ekranu. W tablicach
40      'X1 i Y1 są wszystkie punkty dla pozycji pierwszej. W tab-
50      'licach X2 i Y2 są przechowywane dane dla pozycji drugiej.
60      '***********************************************************
70  CLEARSCREEN
80  DIM X1(15), Y1(15), X2(15), Y2(15)      'XM - maksymalna wartość X
90  XM = 320
100 FOR K = 1 TO 13            'Odczytaj dane dla pozycji pierwszej
110   READ X1(K), Y1(K)
120 NEXT K
130 FOR K = 1 TO 12            'Odczytaj dane dla pozycji drugiej
140   READ X2(K), Y2(K)
150 NEXT K
160 XO = 0           'XO - przesunięcie wzdłuż ekranu
170 GRAPHICS
180 IF XO + X1(12) > XM THEN 800    'Czy pozycja pierwsza na ekranie?
190 GOSUB 270                       'Rysuj pozycję pierwszą
200 GOSUB 390                       'Usuń pozycję pierwszą
210 XO = XO + 15                    'Przesuń o 15 pikseli
220 IF XO + X2(11) > XM THEN 800    'Czy pozycja druga na ekranie?
230 GOSUB 510                       'Rysuj pozycję drugą
240 GOSUB 630                       'Usuń pozycję drugą
250 XO = XO + 20                    'Przesuń o 20 pikseli
260 GOTO 190
```

Program 7.14 (cd.)

```
270 '*********************** Rysuj pozycję pierwszą *********************
280 COLOR 1,0
290 FOR K = 1 TO 3                   'Rysuj jedną nogę i tułów
300   DRAWLINE XO + X1(K), Y1(K) TO XO + X1(K+1), Y1(K+1)
310 NEXT K
320 DRAWLINE XO + X1(5), Y1(5) TO XO + X1(6), Y1(6)  'Rysuj drugą nogę
330 DRAWLINE XO + X1(6), Y1(6) TO XO + X1(7), Y1(7)
340 FOR K = 8 TO 11                                  'Rysuj ręce
350   DRAWLINE XO + X1(K), Y1(K) TO XO + X1(K+1), Y1(K+1)
360 NEXT K
370 CIRCLEPLOT XO + X1(13), Y1(13),10                'Rysuj głowę
380 RETURN
390 '******************* Usuń pozycję pierwszą ************************
400 COLOR 0,0
410 FOR K = 1 TO 3
420   DRAWLINE XO + X1(K), Y1(K) TO XO + X1(K+1), Y1(K+1)
430 NEXT K
440 DRAWLINE XO + X1(5), Y1(5) TO XO + X1(6), Y1(6)
450 DRAWLINE XO + X1(6), Y1(6) TO XO + X1(7), Y1(7)
460 FOR K = 8 TO 11
470   DRAWLINE XO + X1(K), Y1(K) TO XO + X1(K+1), Y1(K+1)
480 NEXT K
490 CIRCLEPLOT XO + X1(13), Y1(13),10
500 RETURN
510 '********************** Rysuj pozycję drugą *************************
520 COLOR 1,0
530 FOR K = 1 TO 2
540   DRAWLINE XO + X2(K), Y2(K) TO XO + X2(K+1), Y2(K+1)
550 NEXT K
560 DRAWLINE XO + X2(4), Y2(4) TO XO + X2(5), Y2(5)
570 DRAWLINE XO + X2(5), Y2(5) TO XO + X2(6), Y2(6)
580 FOR K = 7 TO 10
590   DRAWLINE XO + X2(K), Y2(K) TO XO + X2(K+1), Y2(K+1)
600 NEXT K
610 CIRCLEPLOT XO + X2(12),Y2(12),10
620 RETURN
630 '******************** Usuń pozycję drugą *************************
640 COLOR 0,0
650 FOR K = 1 TO 2
660   DRAWLINE XO + X2(K), Y2(K) TO XO + X2(K+1), Y2(K+1)
670 NEXT K
680 DRAWLINE XO + X2(4), Y2(4) TO XO + X2(5), Y2(5)
690 DRAWLINE XO + X2(5), Y2(5) TO XO + X2(6), Y2(6)
700 FOR K = 7 TO 10
710   DRAWLINE XO + X2(K), Y2(K) TO XO + X2(K+1), Y2(K+1)
720 NEXT K
730 CIRCLEPLOT XO + X2(12),Y2(12),10
740 RETURN
750 '*****************************************************************
760 DATA 14,150,20,133,15,120,20,93,5,145,25,133,15,120
770 DATA 20,115,10,110,19,92,20,108,30,113,20,83
780 DATA 2,132,25,136,40,93,43,150,50,130,30,120
790 DATA 30,111,22,103,38,95,43,110,58,104,40,83
800 END
```

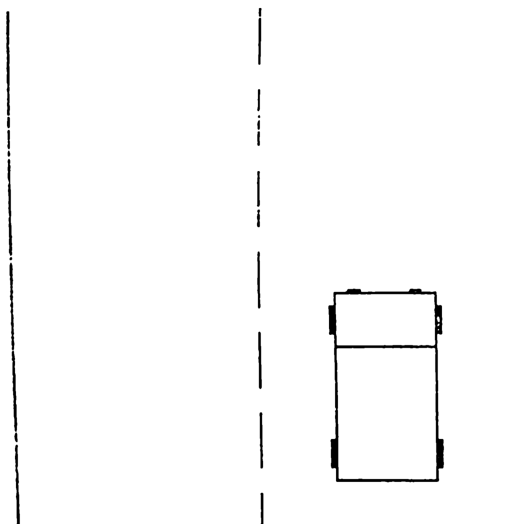


Rys. 7.12. Ruch złożony — biegnąca sylwetka uzyskana za pomocą dwóch ramek (program 7.14)

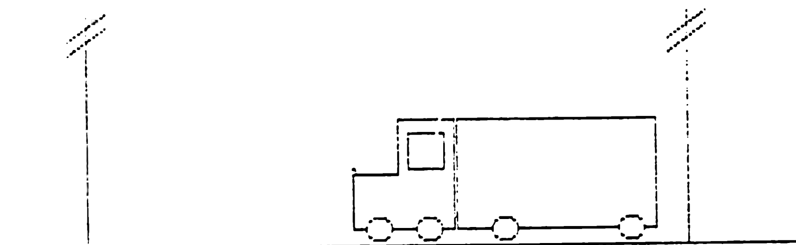
7.4. Ruch tła

Wraz ze wzrostem liczby rysowanych odcinków i obliczeń rośnie czas potrzebny na animację sceny. Dla złożonych obiektów czas animacji może być tak duży, że utraci się wrażenie ruchu, który miał być prezentowany na ekranie. Możemy przyspieszyć ruch dokonując animacji jedynie prostych figur rysowanych z niewielkiej liczby odcinków. Możemy również ograniczyć animację jedynie do prostych rodzajów ruchu. Inna metoda uzyskania wrażenia ruchu polega na poruszaniu tła sceny.

Prosty przykład ruchu tła zilustrowano na rys. 7.13. Przesuwając linię środkową jezdni uzyskamy wrażenie ruchu samochodu. Przesuwanie słupów telefonicznych (rys. 7.14) w prawo daje efekt ruchu ciężarówki w lewo. Słupy telefoniczne mogą być tak rozmieszczone, że nigdy nie zajdzie potrzeba rysowania ich nad ciężarówką. W przeciwnym razie część ciężarówki zostałaby wymazana. Jeżeli mu-



Rys. 7.13. Symulacja ruchu samochodu w górę ekranu uzyskana przez przesuwanie środkowej linii jezdni w dół



Rys. 7.14. Symulacja ruchu ciężarówki w lewo uzyskana dzięki przesuwaniu słupów telefonicznych w prawo

siny przesuwać tło przez obiekt, możemy tak je modyfikować, aby w trakcie animacji w danym punkcie była rysowana tylko część tła albo obiektu. Te testy i modyfikacje zwolnią jednak animację.

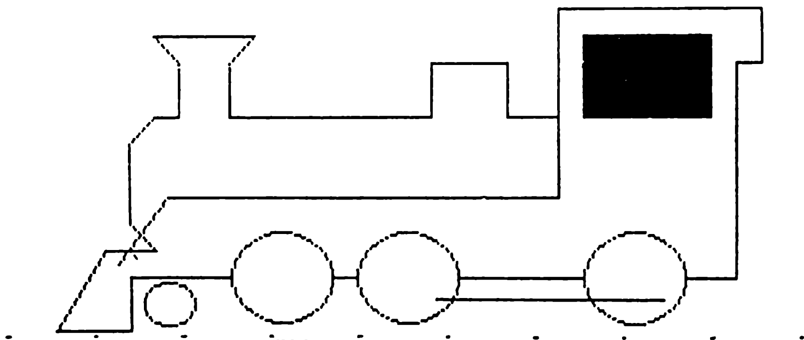
Możliwe są różne rodzaje ruchów tła. W programie 7.15 połączono ruch tła z prostym ruchem obiektu, co symuluje ruch bardziej złożonego obiektu. Wynik pokazano na rys. 7.15. Ruchomy drążek łączący koła lokomotywy i ciąg stale poruszających się podkładów tworzonych z pojedynczych pikseli umożliwiają symulowanie ruchu lokomotywy. Wolniejszy lub szybszy ruch tła uzyskuje się przyjmując mniejszy lub większy odstęp między pozycjami obiektu tła. Możemy również dowolnie zwolnić ruch dobierając czas opóźnienia między ramkami.

Program 7.15. Symulacja ruchu metodą przesuwania tła: lokomotywa z ruchomymi korbowodami i ruchomym torem

```
10   'PROGRAM 7.15. Lokomotywa z ruchomymi korbowodami łączącymi koła i
20        'przesuwającym się torem
30        'Program rysuje lokomotywę i obraca korbowód łączący dwa
40        'tylne koła. Program przesuwa również punkty pod lokomotywą,
45        'co symuluje ruch po torze.
50        '**********************************************************
60   CLEARSRCEEN
70   YA = 5/6                     YA - korekcja rozdzielczości
80   GRAPHICS
90        'Rysuj lokomotywę
100  READ X1,Y1
110  FOR K = 1 TO 25
120    READ X2,Y2
130    DRAWLINE X1,Y1 TO X2,Y2
140    X1 = X2
150    Y1 = Y2
160  NEXT K
170       'Dodaj okno
180  READ XL, XR, YT, YB
190  FOR X = XL TO XR
200    DRAWLINE X,YT TO X,YB
210  NEXT X
220       'Dodaj inne detale
230  FOR K = 1 TO 6
240    READ X1, Y1, X2, Y2
250    DRAWLINE X1,Y1 TO X2,Y2                  'Ostatnie wartości określają
260  NEXT K                                             'końce korbowodu
```

Program 7.15 (cd.)

```
270       'Dodaj koła
280 FOR K = 1 TO 4
290   READ XC, YC, R
300   CIRCLEPLOT XC,YC,R                  'Ostatnie wartości XC i YC do-
310 NEXT K                                     'tyczą tylnego koła
320       '
330       '*********** Obracaj korbowód, przesuwaj podkłady ***********
340 Y = 152           'Wartość Y dla torów
350 R = 15            'R - promień obrotu korbowodu
360 RE = 360 * 3.14159 / 180      'Zamiana 360 i 50 stopni na radiany
370 DA = 50  * 3.14159 / 180
380 FOR A = DA TO RE STEP DA
390   X1 = XC + R * SIN(A)
400   Y1 = YC + R * COS(A) * YA
410   COLOR 1,0
420   DRAWLINE X1,Y1 TO X1-90,Y1  'Rysuj nową pozycję korbowodu (długość
430   FOR X = XS TO 319 STEP 35   '90 pikseli). Rysuj podkłady co 35
440     POINTPLOT X,Y             'pikseli
450   NEXT X
460   FOR X = XS TO 319 STEP 35   'Usuń podkłady
470     POINTOFF X,Y
480   NEXT X
490   XS = XS +7                  'Następny zestaw punktów będzie prze-
500   IF XS >=30 THEN XS = 0      'sunięty o 7 pikseli. Ponieważ XS = 0,
510   COLOR 0,0                   'to punkty przychodzą z lewej strony
520   DRAWLINE X1,Y1 TO X1-90,Y1  'Usuń bieżącą pozycję korbowodu
530 NEXT A
540 GOTO 380
550 '*******************************************************************
560       'Kontur
570 DATA 270,130,290,130,290,50,300,50,300,30,220,30,220,70,200,70
580 DATA 200,50,170,50,170,70,90,70,90,50,100,40,60,40,70,50,70,70
590 DATA 60,70,50,80,50,110,60,120,40,120,20,150,50,150,50,130,90,130
600       'Okno
610 DATA 230,280,40,70
620       'Inne detale
630 DATA 220,70,220,100,220,100,65,100
640 DATA 65,100,45,125,130,130,140,130
650 DATA 0,154,319,154,180,130,230,130
660       'Koła
670 DATA 65,140,10,110,130,20,160,130,20,250,130,20
680 END
```



Rys. 7.15. Symulacja ruchu lokomotywy w lewo uzyskana przez ruch korbowodu i ruch podkładów toru (program 7.15)

ZADANIA

7.1. Napisz program ruchu piksela (lub okręgu) między kilkoma pionowymi i poziomymi granicami. Pozycję i długość każdej granicy można wybrać losowo. Poruszaj punkt z przyrostami jednostkowymi w obu kierunkach pionowym i poziomym tak, aby tor ruchu był linią ukośną. Gdy punkt napotyka pionową granicę, zmienia się znak przyrostu X, gdy osiągnie poziomą granicę, zmienia się znak przyrostu Y. W celu sprawdzenia, czy punkt zbliża się do losowo umieszczonej granicy, przed przesunięciem punktu użyj funkcji testującej, czy w następnej pozycji punkt znajdzie się na granicy.

7.2. Zmodyfikuj program z zadania 7.1 tak, aby wyświetlał punkt (lub piłkę) w różnych kolorach w trakcie krążenia po ekranie. Inne możliwe modyfikacje polegają na zmianie kolorów tylko w czasie odbicia, zostawianiu śladu punktu (zostanie wyświetlony kolorowy wzór) lub generowaniu dźwięku przy odbiciu od granicy.

7.3. Zmodyfikuj program z zadania 7.1 tak, aby punkt poruszał się szybciej. Zachowując równe wartości przyrostów poziomego i pionowego przyjmij wartości większe niż 1. Teraz, każdorazowo przy określaniu kolejnej pozycji trzeba sprawdzić, czy między bieżącą a następną pozycją punktu nie ma granicy.

7.4. Napisz program odbijania piksela (lub kółka) tam i z powrotem między dwoma określonymi punktami (X1, Y1) i (X2, Y2). Wartości przyrostów (DX i DY) wybiera się tak, aby stosunek DY/DX był równy nachyleniu toru. Uzyskuje się to dzieląc przedziały X2 — — X1 i Y2 — Y1 przez 10 lub inny dogodny współczynnik.

7.5. Napisz program tworzenia labiryntu wewnątrz prostokąta z programu 7.3. Umieść kółko w pewnym położeniu początkowym i umożliw interakcyjne określanie kierunku ruchu. Ruch może być ograniczony do ścieżek poziomych i pionowych. Zaprogramuj ruch kółka z krokiem jednostkowym w kierunku zmienianym przez urządzenie wejściowe lub w czasie zderzenia kółka ze ścianą. Zderzenia można określić za pomocą funkcji, która wskazuje, czy pozycja piksela jest na ekranie, czy poza nim. Ruch kończy się z chwilą, gdy kółko dotrze do wyjścia z labiryntu. W celu przekształcenia tego programu w grę odejmuj punkty za zderzenia i drukuj końcowy wynik.

7.6. Zmodyfikuj program 7.3 tak, aby można było ustalać początkowe wartości przyrostów DX i DY (DX różne od DY). Przyspiesz ruch zwiększając przyrosty o 5 po każdym trzecim odbiciu od lewej ściany. Ruch można zatrzymać, gdy jeden z przyrostów przekroczy wartość 40.

7.7. Napisz program wyświetlania piłki (lub innego obiektu), która porusza się po ekranie wzdłuż toru wyznaczonego przez określoną krzywą. Krzywa może być opisana: funkcją SIN, wielomianem czwartego stopnia lub inną funkcją, która będzie symulować ruch w górę i w dół.

7.8. Napisz program wyświetlania punktu (lub piłki), który porusza się wzdłuż spirali zaczynającej się w środku ekranu. Zmieniając kolor i pomijając usuwanie punktów można wypełnić ekran kolorową spiralą. Zmieniając kierunek ruchu, gdy punkt osiągnie brzeg ekranu, można uzyskać efekt ruchu po spirali tam i z powrotem.

7.9. Napisz program animacji okręgu umieszczonego w środku ekranu korzystając z metody skalowania. Zwiększaj okrąg dopóty, dopóki nie osiągnie brzegów ekranu, a następnie zacznij zmniejszanie okręgu. Kontynuuj proces zmieniając każdorazowo kolor przy przejściu z fazy zmniejszania do fazy zwiększania okręgu i odwrotnie.

7.10. Napisz program symulacji ruchu obiektu metodą poruszania tła. Przykładowymi obiektami mogą być: pociąg, statek z dymem ulatującym z komina, sceny z rys. 7.13 lub 7.14.

7.11. Napisz program wyświetlania tarczy zegara w środku ekranu. Narysuj wskazówki jako strzały (dla godzin i minut) i obracaj je tak, aby mniejsza wskazówka obracała się o 30°

w czasie, gdy duża robi pełny obrót (360°). Można dodać wskazówkę sekundową, która obraca się o 360° w czasie, gdy wskazówka minutowa obraca się o 6°.

7.12. Napisz program poruszania odcinka wzdłuż następujących krzywych: a) elipsy; b) wielomianu trzeciego stopnia; c) sinusoidy. Odcinek ma być stale styczny do krzywej.

7.13. Napisz program wyświetlania samolotu lub statku kosmicznego wykonującego pętle wokół okręgu.

8. Okna obrazowe

Obrazy graficzne mogą być przetwarzane wieloma sposobami. Omawialiśmy już metody zmiany wymiarów obrazu i przenoszenia figur z jednej części ekranu do innej lub zmiany ich orientacji. Te transformacje mogą być stosowane do całego obrazu lub do wskazanych obszarów w obrazie. Możliwe jest także wskazywanie obszarów obrazu dla innych typów modyfikacji. Te obszary mogą być wyróżnione, usunięte z obrazu lub też mogą być zachowane i przekształcone, podczas gdy reszta obrazu jest usuwana.

8.1. Wyróżnianie

Często trzeba nadać specjalne znaczenie pewnym obszarom obrazu, aby zwrócić na nie uwagę. Obszar wyróżniony może być narysowany w jasnym kolorze lub przy zwiększonej jasności. W niektórych systemach przewiduje się opcjonalną „dużą jasność" obrazu. W wyniku jej stosowania linie lub tekst wybranego obszaru są wyświetlane jaśniej niż na innych częściach ekranu. Podobny efekt osiąga się rysując linie o podwójnej grubości w obszarze wymagającym wyróżnienia: uzyskuje się to przez wyświetlanie dwóch linii w odstępie jednostkowym zamiast jednej linii. Inna metoda wyróżniania polega na migotaniu części obrazu. Można także zwrócić uwagę na wybrany obszar ekranu rysując wokół niego prostokąt lub okrąg (rys. 8.1).

W celu nałożenia okręgu na obraz wybieramy położenie jego środka (XC, YC) i promień R. Za każdym razem gdy chcemy wyróżnić tę część ekranu, wywołujemy podprogram rysowania okręgu lub rozkaz CIRCLEPLOT. Prosty program umożliwia eksperymentowanie z różnymi położeniami i wymiarami okręgu w celu uzyskania żądanego efektu (program 8.1).

W przypadku prostokąta wystarczy określić współrzędne dwóch przeciwległych wierzchołków. Można także obmyślić program, który zapyta użytkownika,

1. RAMA

2. SIODEŁKO

3. KOŁA

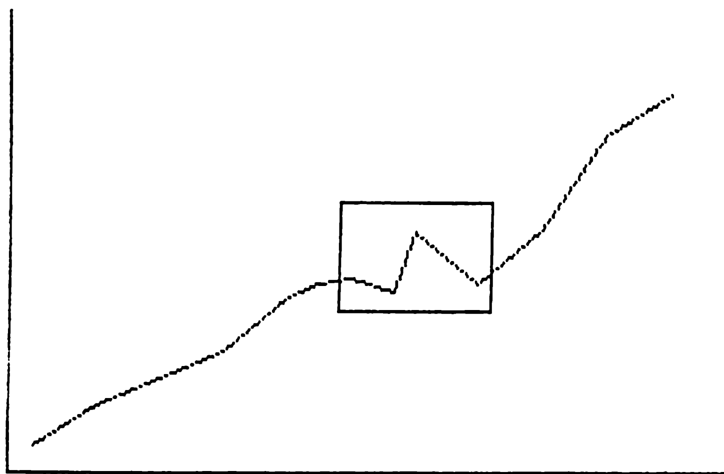
4. HAMULCE

5. PEDAŁY

6. WIDELEC

7. KIEROWNICA

(a)



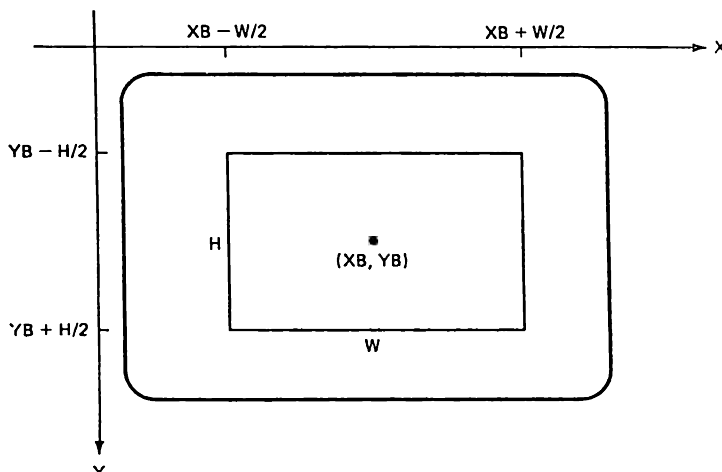
(b)

Rys. 8.1. Wyróżnianie części obrazu za pomocą okręgu (a) i prostokąta (b)

Program 8. 1. Wyróżnianie za pomocą okręgu

```
10  'PROGRAM 8.1. Wyróżnianie za pomocą okręgu
20  CLEARSCREEN
30  DIM X(15), Y(15)
40  XM = 299
50  YM = 199
60      '******** Odczytaj dane do rysunku **************
70      '
80      '
90      '
100     '
110     '
120 GOSUB 160        'Narysuj obraz
130 GOSUB 250        'Narysuj wyróżnienie w postaci okręgu
140 GOTO 600
150     '
160     '*************** Program rysowania **************
170 CLEARSCREEN
180 GRAPHICS
190     '
200     '
210     '
220     '
230 RETURN
240     '
250     '********* Dodaj wyróżnienie w postaci okręgu ********
260 POSITION 2,1
270 PRINT "Srodek okręgu i promień";
280 INPUT XC, YC, R
290 IF XC+R<=XM AND XC-R>=0 AND YC+R<=YM AND YC-R>=0 THEN 340
300 POSITION 1,1
310 PRINT "Okrąg poza ekranem"
320 PRINT "                                        "
330 GOTO 260
340 CIRCLEPLOT XC,YC,R
350 POSITION 1,1
360 PRINT "                                        "
370 POSITION 2,1
380 PRINT "                                        "
390 POSITION 2,1
400 PRINT "Napisz K aby skończyć, Z aby zmienić okrąg";
410 INPUT M$
420 POSITION 2,1
430 PRINT "                                        "
440 IF M$ = "K" THEN 580
450     'Wymaż aktualną pozycję
460 COLOR 0,0
470 CIRCLEPLOT XC,YC,R
480 COLOR 1,0
490     'Czy wymazywanie niszczy obraz?
500 POSITION 2,1
510 PRINT "Czy chcesz narysować powtórnie (T/N)?";
520 INPUT D$
530 POSITION 2,1
540 PRINT "                                        "
550 IF D$ = "N" THEN 250
560 GOSUB 160
570 GOTO 250
580 RETURN
590     '************************************************************
600 END
```

gdzie umieścić środek prostokątnego pudełka i jak duże ma ono być (rys. 8.2). Przykładowy program 8.2 rysuje prostokąt o współrzędnych środka (XB, YB), wysokości H i szerokości W.



Rys. 8.2. Współrzędne wierzchołków prostokąta można określić znając jego szerokość (W), wysokość (H) i położenie środka (XB, YB)

Program 8.2. Wyróżnianie za pomocą prostokąta

```
10   'PROGRAM 8.2. Wyróżnianie za pomocą prostokąta
20   CLEARSCREEN
30   DIM X(15), Y(15)
40   XM = 279
50   YM = 159
60       '************* Odczytaj dane do rysunku ***************
70       '
80       '
90       '
100      '
110      '
120  GOSUB 160              'Narysuj obraz
130  GOSUB 260              'Dodaj wyróżnienie w postaci prostokąta
140  GOTO 770
150      '
160      '**************** Program rysowania *****************
170  CLEARSCREEN
180  GRAPHICS
190      '
200      '
210      '
220      '
230      '
240  RETURN
250      '
260      '********** Dodaj wyróżnienie w postaci prostokąta **********
270  POSITION 2,1
```

Program 8.2 (cd.)

```
280 PRINT "Podaj współrzędne środka prostokąta (X i Y)";
290 INPUT XB, YB
300 POSITION 2,1
310 PRINT "                                    "
320 POSITION 2,1
330 PRINT "Szerokość i wysokość prostokąta";
340 INPUT BW, BH
350     'Określ boki prostokąta (lewy, prawy, góra, dół)
360 L = XB - BW / 2
370 R = XB + BW / 2
380 T = YB - BH / 2
390 B = YB + BH / 2
400 IF L<R AND L>=0 AND R<=XM AND T<B AND T>=0 AND B<=YM THEN 450
410 POSITION 1,1
420 PRINT "Prostokąt poza ekranem"
430 PRINT "                                    "
440 GOTO 270
450 DRAWLINE L,T TO L,B
460 DRAWLINE L,B TO R,B
470 DRAWLINE R,B TO R,T
480 DRAWLINE R,T TO L,T
490 POSITION 1,1
500 PRINT "                                    "
510 POSITION 2,1
520 PRINT "                                    "
530 POSITION 2,1
540 PRINT " Napisz K aby skończyć, Z aby zmienić prostokąt"
550 INPUT M$
560 POSITION 2,1
570 PRINT "                                    "
580 IF M$ = "K" THEN 750
590     'Wymaż aktualną pozycję prostokąta
600 COLOR 0,0
610 DRAWLINE L,T TO L,B
620 DRAWLINE L.B TO R,B
630 DRAWLINE R,B TO R,T
640 DRAWLINE R,T TO L,T
650 COLOR 1,0
660     'Czy wymazywanie prostokąta niszczy obraz?
670 POSITION 2,1
680 PRINT "Narysować obraz powtórnie (T/N)?";
690 INPUT D$
700 POSITION 2,1
710 PRINT "                                    "
720 IF D$ = "N" THEN 260
730 GOSUB 160
740 GOTO 260
750 RETURN
760     '**********************************************************
770 END
```

Wyróżnianie części obrazu za pomocą okręgu lub prostokąta jest bardzo użyteczne i stosowane w sprawozdaniach lub na wykładach. Obraz na ekranie z wyróżnieniem w określonych miejscach może być wyprowadzony na drukarkę przy każdej zmianie położenia wyróżnienia. Wydruki mogą stanowić część sprawozdania. Dla celów prezentacji na ekranie można zrealizować ruchome wyróżnienia. Będą one zmieniać położenia tak, aby pokrywać tę część obrazu, która jest

właśnie analizowana. Prosta pętla czasowa umożliwia utrzymanie wyróżnienia na ekranie w jednym miejscu do czasu, gdy prelegent będzie gotów do analizy następnej części rysunku. Można w ten sposób wyróżniać listę obiektów, części wykresu lub rysunku.

8.2. Wymazywanie i obcinanie

Zamiast wyróżnienia części obszaru może zajść potrzeba usunięcia części złożonego obrazu graficznego. Za pomocą wyróżnienia określa się obszary do wymazania lub obcięcia. W pierwszym przypadku usuwamy wybrany obszar, w drugim – całe otoczenie wybranego obszaru, który nie ulega zmianie.

Wymazywanie

Metodę wyróżniania za pomocą okręgu lub prostokąta stosuje się do wymazywania. Decyzję o usunięciu pewnej części gotowego obrazu graficznego możemy podjąć z wielu powodów. Możemy, na przykład, chcieć uprościć złożony obraz lub zrobić miejsce dla dalszych części obrazu. Możemy też wypróbować różne formy ułożenia rysunku lub eksperymentować z efektami wizualnymi. Możemy także potrzebować miejsca do powiększenia części rysunku lub wykonania innych operacji na nim.

Część rysunku można usunąć przez taką zmianę programu rysowania, aby przy powtórnym rysowaniu zbędne części nie pojawiły się. Może to wymagać znacznej zmiany programu wyświetlania graficznego. Gdybyśmy w pewnej chwili chcieli mieć z powrotem usunięte części, to znów musielibyśmy zmieniać program. Jeśli takie operacje są częste, to celowe jest opracowanie uniwersalnego programu wymazywania części rysunku w zadanym obszarze.

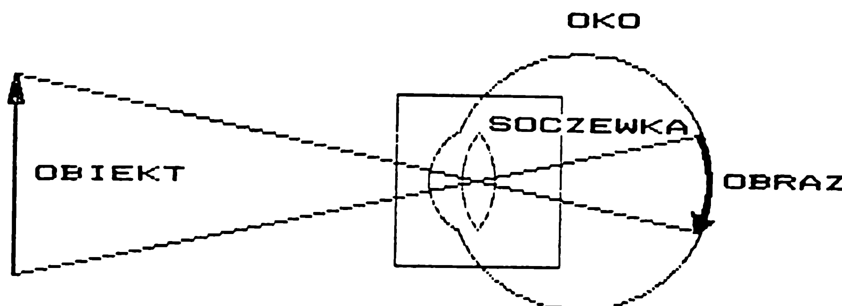
Określenie obszaru do wymazania za pomocą prostokąta jest prostym sposobem określenia znaków, pikseli i odcinków do usunięcia. Wystarczy po prostu wypełnić ten prostokątny obszar liniami o tym samym kolorze co tło. Program, który to wykona, będzie potrzebował na wejściu danych o kolorze tła oraz o wymiarach i położeniu prostokąta. Jeśli obraz nie jest kolorowy, to instrukcja POINT-OFF (usuń punkt) może posłużyć do wymazania każdego piksela wewnątrz prostokąta.

Można także wymazywać wewnątrz obszaru o kształcie okręgu, będzie to jednak wolniejszy proces, gdyż program będzie musiał obliczać położenie brzegów obszaru. Te punkty będą służyć jako współrzędne końców odcinków wrysowanych do wnętrza obszaru. W niektórych systemach przewidziano możliwość podania jako parametru koloru wnętrza okrągłego lub prostokątnego obszaru. Wnętrze takiej figury jest automatycznie zapełniane żądanym kolorem w trakcie jej rysowania.

Obcinanie

W niektórych przypadkach chcielibyśmy zachować pewien obszar obrazu i usunąć całą resztę. Takimi przypadkami będą: potrzeba powiększenia małej części obrazu lub wykresu do wymiarów ekranu albo potrzeba oddzielnego przedstawienia rysunku będącego częścią składową większego rysunku w celu prezentacji lub włączenia go do raportu. Obszar do zapamiętania wybieramy za pomocą prostokątnego obszaru zwanego *oknem*. Wymazanie pozostałych części poza obszarem okna jest nazywane *obcinaniem*.

Program wykonujący obcinanie musi zidentyfikować i zapamiętać tekst, niepołączone piksele i odcinki linii prostych z obszaru okna. Na rysunku 8.3 poka-

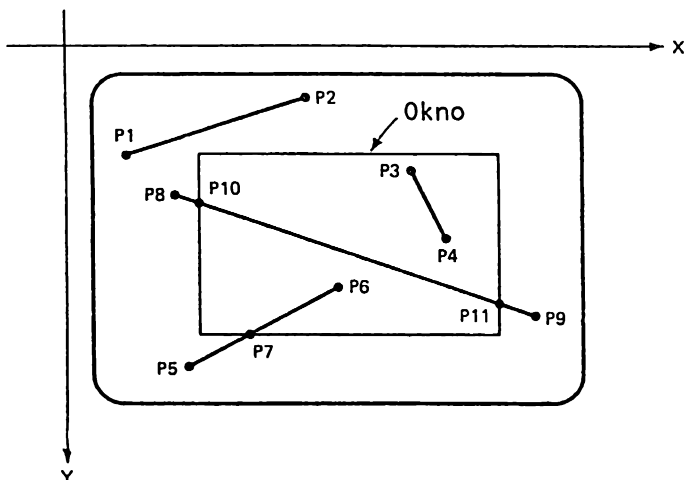


Rys. 8.3. Okno wokół tej części obrazu, która ma być zachowana przez program obcinający

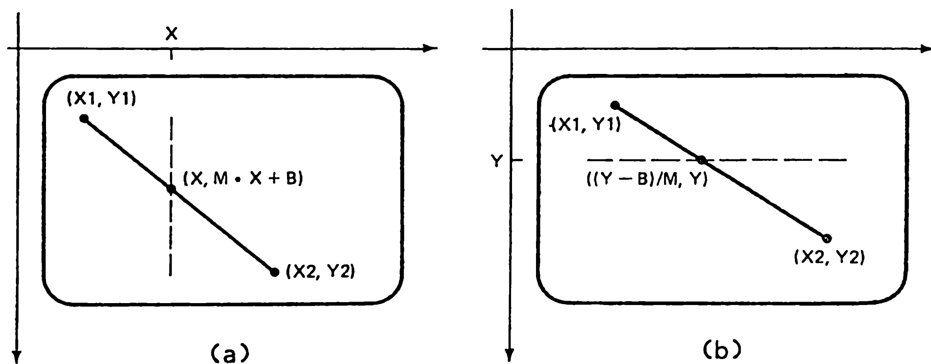
zano okno wyróżniające część obrazu. Wszystko, co znajduje się w obszarze tego okna, powinno być zachowane; wszystko, co jest na zewnątrz, ma być obcięte. Program obcinający powinien sprawdzić współrzędne końców linii i określić zapamiętywany tekst.

Rozważymy najpierw projekt programu obcinania linii i punktów. Do programu będą wprowadzane: współrzędne lewego górnego rogu okna jako (XW, YW), wysokość i szerokość okna odpowiednio jako HW i WW. Współrzędne obrazu będą zapamiętane w tablicach X i Y. Wszelkie współrzędne obrazu zapamiętane przez program obcinania zostaną umieszczone w nowych tablicach.

Izolowany punkt (jedna cząstka linii) będzie zapamiętany przez program obcinania, jeśli znajduje się on w obszarze okna, to znaczy jeśli jego współrzędna X jest zawarta między XW a XW + WW, a współrzędna Y między YW a YW + HW. Odcinek będzie zapamiętany w całości, jeśli obydwa jego końce leżą wewnątrz okna. Odcinek będzie w całości wymazany, jeśli cały leży poza oknem. Jeśli część odcinka leży poza oknem, to tę część obetniemy, a zachowamy tę część, która leży wewnątrz okna. Na rysunku 8.4 pokazano różne położenia odcinków względem okna. Punkty przecięcia z poziomymi i pionowymi granicami okna oblicza się na podstawie współrzędnych okna i równań linii (nachylenia M oraz przesunięcia B) — rys. 8.5.



Rys. 8.4. Linia o koń ach w punktach P1 i P2 (obydwa punkty znajdują się na zewnątrz okna) zostanie wymazana p zez program obcinania. Linia o końcach w punktach P3 i P4 (obydwa wewnątrz okna) będzie zachowana. Procedura obcinania zachowa także odcinki linii od punktu P6 do P0 i od P17 do P11



Rys. 8.5. Linia o nachyleniu M i przesunięciu B przecięta pionową i poziomą granicą okna: a) punkt przecięcia pionowej granicy okna o współrzędnych (X, M * X + B); b) punkt przecięcia poziomej granicy okna o współrzędnych (Y — B)/M, Y)

W programie 8.3 rozpoczynamy sprawdzanie współrzędnych od lewego brzegu okna. Punkty o współrzędnej X większej niż wartość X dla lewego brzegu okna są zapamiętywane w tablicach X1 i Y1. Punkty przecięcia linii z lewym brzegiem okna są także zapamiętywane w tablicach X1 i Y1. Następnie punkty z tablic X1 i Y1 są obcinane w stosunku do górnego brzegu okna. Wszystkie punkty przecięcia i dowolne punkty o wartości Y większej niż współrzędne Y tego brzegu są zapamiętywane w tablicach X2 i Y2. Te punkty są znów obcinane w stosunku do

Program 8.3. Obcinanie punktów i linii rysunku samolotu

```
10  'PROGRAM 8.3. Obcinanie poza oknem
20      'Program rysuje figurę na podstawie danych punktów w tablicach
30      'X i Y. Wybiera się okno. Linie i punkty poza oknem są
40      'obcinane.  Obcinanie  następuje w stosunku  do  każdego
50      'brzegu okna, w kolejności: lewy, góra, prawy, dół. Przy
60      'obcinaniu punkty z tablic X i Y są obcinane w stosunku do
70      'lewego brzegu; punkty wewnątrz okna są zapamiętywane w X1, Y1.
80      'Te punkty są następnie obcinane w stosunku do górnego brzegu.
90      'Punkty wewnątrz okna są zapamiętywane w X2, Y2. Punkty z
100     'X2, Y2 są obcinane  w stosunku do prawego brzegu, a tablice
110     'X1, Y1 są używane do pamiętania punktów wewnątrz okna. W końcu
120     'te punkty są obcinane w stosunku do dolnego brzegu; punkty
130     'wewnętrzne zapamiętuje się w X2, Y2. Wymazuje się ekran, potem
140     'zaś rysuje się okno i część rysunku, która jest wewnątrz okna.
150     '***************************************************
160 CLEARSCREEN
170 DIM X(8,20), Y(8,20), X1(8,20), Y1(8,20), X2(8,20), Y2(8,20)
180 XM = 319
190 YM = 199          'XM i YM są maksymalnymi wartościami w systemie
200     '************ Odczytaj dane do rysunku ****************
210 READ N                      'N jest liczbą części rysunku
220 FOR P = 1 TO N
230   READ NE(P)                'Liczba elementów w części P
240   FOR E = 1 TO NE(P)
250     READ X(P,E), Y(P,E)
260   NEXT E
270 NEXT P
280 GRAPHICS
290 GOSUB 340                   'Narysuj obraz
300 GOSUB 450                   'Wykreśl okno
310 GOSUB 850                   'Obcinaj poza oknem
320 GOSUB 2480                  'Narysuj obcięte punkty
330 GOTO 2470
340     '***************** Program rysowania *********************
350 CLEARSCREEN
360 FOR P = 1 TO N
370   IF NE(P) > 1 THEN 400     'Więcej niż jeden punkt
380   POINTPLOT X(P,1), Y(P,1)  'Tylko jeden punkt w tej części
390   GOTO 430                  'Przejdź do następnej części
400   FOR E = 1 TO NE(P) - 1
410     DRAWLINE X(P,E), Y(P,E) TO X(P,E+1), Y(P,E+1)
420   NEXT E
430 NEXT P
440 RETURN
450     '********************* Wykreśl okno ***********************
460 POSITION 1,1
470 PRINT "Górny lewy róg okna";
480 INPUT XW, YW
490 POSITION 1,1
```

Program 8.3 (cd.)

```
500 PRINT "                                          ";
510 POSITION 1,1
520 PRINT "Szerokość i wysokość okna";
530 INPUT WW, HW
540 POSITION 1,1
550 PRINT "                                          ";
560 L = XW
570 R = XW + WW
580 T = YW
590 B = YW + HW
600 IF L<R AND L>=0 AND R <=XM AND T<B AND T>=0 AND B<=YM THEN 660
610 POSITION 1,1
620 PRINT "Okno poza ekranem. Spróbuj jeszcze raz"
630 POSITION 1,1
640 PRINT "                                          ";
650 GOTO 460
660 DRAWLINE L,T TO L,B
670 DRAWLINE L,B TO R,B
680 DRAWLINE R,B TO R,T
690 DRAWLINE R,T TO L,T
700 POSITION 1,1
710 PRINT "Napisz K aby kontynuować, Z aby zmienić okno";
720 INPUT M$
730 IF M$ = "K" THEN 840
740     'Wymaż aktualną pozycję
750 POSITION 1,1
760 PRINT "                                          ";
770 COLOR 0,0
780 DRAWLINE L,T TO L,B
790 DRAWLINE L,B TO R,B
800 DRAWLINE R,B TO R,T
810 DRAWLINE R,T TO L,T
820 COLOR 1,0
830 GOTO 450
840 RETURN
850     '****************** Program obcinania *********************
860     'Obcinanie punktów z tablic X, Y w stosunku do lewego brzegu;
870     ' wyniki zapamiętuje się w tablicach X1, Y1.
880 P1 = 1
890 FOR P = 1 TO N
900    E1 = 0
910    FOR E = 1 TO NE(P) - 1
920      IF X(P,E) >= L THEN 940
930      IF X(P,E) <  L THEN 1020
940      'Pierwszy punkt linii jest wewnętrzny
950      E1 = E1 + 1
960      X1(P1,E1) = X(P,E)
970      Y1(P1,E1) = Y(P,E)
980      'Co z drugim końcem?
990      IF  X(P,E+1) < L THEN GOSUB  1190  'Jeśli zewnętrzny, znajdź
1000     'przecięcie. Jeśli nie, kontynuuj
1010     GOTO 1070
1020     '-----------------------------------------------
1030     'Pierwszy punkt jest na zewnątrz
1040     'Co z drugim?
1050     IF X(P,E+1) >= L THEN GOSUB 1190   'Jeśli wewnętrzny, znajdź
1060     'przecięcie. Jeśli nie, kontynuuj
1070    NEXT E
1080    'Zatroszcz się o ostatni punkt
1090    IF X(P,NE(P)) < L THEN 1130
1100    E1 = E1 + 1
1110    X1(P1,E1) = X(P,NE(P))
1120    Y1(P1,E1) = Y(P,NE(P))
1130    IF E1 = 0 THEN 1160        'Nie ma elementów w oknie
```

Program 8.3 (cd.)

```
1140   ME(P1) = E1                  'Jeśli są, to zapamiętaj ich liczbę
1150   P1 = P1 + 1                  'Przejdź do następnej części
1160 NEXT P
1170 MN = P1 -1                     'MN jest liczbą części na prawo od L
1180 GOTO 1260
1190    '^^^^^^^^^^^^^ Program znajdowania przecięć ^^^^^^^^^^^^^^
1200 E1 = E1 + 1
1210 M = (Y(P,E+1) - Y(P,E)) / (X(P,E+1) - X(P,E))
1220 Y1(P1,E1) = M * (L - X(P,E)) + Y(P,E)
1230 X1(P1,E1) = L
1240 RETURN
1250    '**************************************************************
1260    'Obetnij  punkty  w  X1,  Y1  w  stosunku  do górnego brzegu
1265    'Zapamiętaj punkty wewnętrzne w tablicach X2, Y2
1270 P1 = 1
1280 FOR P = 1 TO MN
1290   E1 = 0
1300   FOR E = 1 TO ME(P) - 1
1310     IF Y1(P,E) >= T THEN 1330
1320     IF Y1(P,E) <  T THEN 1420
1330     'Pierwszy punkt linii jest wewnętrzny
1340     E1 = E1 + 1
1350     Y2(P1,E1) = Y1(P,E)
1360     X2(P1,E1) = X1(P,E)
1370     'Co z drugim punktem?
1380     IF Y1(P,E+1) < T THEN GOSUB 1580       'Jeśli zewnętrzny,
1390     'to znajdź przecięcie. Jeśli nie, kontynuuj
1400     GOTO 1460
1410     '------------------------------------------
1420     'Pierwszy punkt jest na zewnątrz
1430     'Co z drugim punktem?
1440     IF Y1(P,E+1) >= T THEN GOSUB 1580     'Jeśli wewnętrzny,
1450     'to znajdź przecięcie. Jeśli nie, kontynuuj
1460   NEXT E
1470   'Zatroszcz się o ostatni punkt
1480   IF Y1(P,ME(P)) < T THEN 1520
1490   E1 = E1 +1
1500   Y2(P1,E1) = Y1(P,ME(P))
1510   X2(P1,E1) = X1(P,ME(P))
1520   IF E1 = 0 THEN 1550                          'Brak elementów w oknie
1530   ME(P1) = E1                 'Zapamiętaj liczbę elementów w oknie
1540   P1 = P1 + 1                 'Przejdź do następnej części
1550 NEXT P
1560 MN = P1 - 1       'MN jest liczbą części na prawo od L, w dół od T
1570 GOTO 1680         'Przejdź do obcinania w stosunku do prawego brzegu
1580    '^^^^^^^^^^^ Program znajdowania przecięć ^^^^^^^^^^^^^^^
1590 E1 = E1 + 1
1600 IF X1(P,E+1) <> X1(P,E) THEN 1630
1610 X2(P1,E1) = X1(P,E)        'Linia pionowa
1620 GOTO 1650
1630 M = (Y1(P,E+1) - Y1(P,E)) / (X1(P,E+1) - X1(P,E))
1640 X2(P1,E1) = ((T - Y1(P,E)) / M + X1(P,E)
1650 Y2(P1,E1) = T
1660 RETURN
1670    '*****************************************************
1680    'Obetnij punkty w X2,Y2 w stosunku do prawego brzegu; wyniki
1690 P1 = 1                                        'zapamiętaj w X1,Y1
1700 FOR P = 1 TO MN
1710   E1 = 0
1720   FOR E = 1 TO ME(P) - 1
1730     IF X2(P,E) <= R THEN 1750
1740     IF X2(P,E) >  R THEN 1840
1750     'Pierwszy punkt jest wewnętrzny
1760     E1 = E1 + 1
```

Program 8.3 (cd.)

```
1770     X1(P1,E1) = X2(P,E)
1780     Y1(P1,E1) = Y2(P,E)
1790     'Co z drugim punktem?
1800     IF X2(P,E+1) > R THEN GOSUB 2000    'Jeśli na zewnątrz, to
1810     'znajdź przecięcie. Jeśli nie, kontynuuj
1820     GOTO 1880
1830     '---------------------------------------
1840     'Pierwszy punkt jest na zewnątrz
1850     'Co z drugim punktem?
1860     IF X2(P,E+1) <= R THEN GOSUB 2000   'Jeśli na zewnątrz, to
1870     'znajdź przecięcie. Jeśli nie, kontynuuj
1880   NEXT E
1890   'Zatroszcz się o ostatni punkt
1900   IF X2(P,ME(P)) > R THEN 1940
1910   E1 = E1 + 1
1920   X1(P1,E1) = X2(P,ME(P))
1930   Y1(P1,E1) = Y2(P,ME(P))
1940   IF E1 = 0 THEN 1970                    'Brak elementów w oknie
1950   ME(P1) = E1               'Zapamiętaj liczbę elementów w oknie
1960   P1 = P1 + 1               'Przejdź do następnej części
1970 NEXT P
1980 MN = P1 - 1  'MN jest l. cz. na prawo od L, w dół od T, w lewo od R
1990 GOTO 2070                 'Obetnij w stosunku do dolnego brzegu
2000   '^^^^^^^^^^^^ Program znajdowania przecięć ^^^^^^^^^^^^
2010 E1 = E1 + 1
2020 M = (Y2(P,E+1) - Y2(P,E)) / (X2(P,E+1) - X2(P,E))
2030 Y1(P1,E1) = M * (R - X2(P,E)) + Y2(P,E)
2040 X1(P1,E1) = R
2050 RETURN
2060   '**************************************************
2070   'Obetnij punkty w X1,Y1  w stosunku do dolnego brzegu;
2080 P1 = 1                          'wyniki zapamiętaj w X2,Y2
2090 FOR P = 1 TO MN
2100   E1 = 0
2110   FOR E = 1 TO ME(P) - 1
2120     IF Y1(P,E) <= B THEN 2140
2130     IF Y1(P,E) >  B THEN 2230
2140     'Pierwszy punkt jest wewnątrz
2150     E1 = E1 + 1
2160     Y2(P1,E1) = Y1(P,E)
2170     X2(P1,E1) = X1(P,E)
2180     'Co z drugim punktem?
2190     IF Y1(P,E+1) > B THEN GOSUB 2390  'Jeśli na zewnątrz,to
2200     'znajdź przecięcie. Jeśli nie, kontynuuj
2210     GOTO 2270
2220     '---------------------------------------
2230     'Pierwszy punkt jest na zewnątrz
2240     'Co z drugim punktem?
2250     IF Y1(P,E+1) <= B THEN GOSUB 2390    'Jeśli wewnątrz, to
2260     'znajdź przecięcie. Jeśli nie, kontynuuj
2270   NEXT E
2280   'Zatroszcz się o ostatni punkt
2290   IF Y1(P,ME(P)) > B THEN 2330
2300   E1 = E1 + 1
2310   Y2(P1,E1) = Y1(P,ME(P))
2320   X2(P1,E1) = X1(P,ME(P))
2330   IF E1 = 0 THEN 2360                  'Brak elementów w oknie
2340   ME(P1) = E1              'Zapamiętaj liczbę elementów w oknie
2350   P1 = P1 +1               'Przejdź do następnej części
2360 NEXT P
2370 MN = P1 -1                 'MN jest liczbą punktów wewnątrz okna
2380 RETURN                     'Koniec programu obcinania
```

Program 8.3 (cd.)

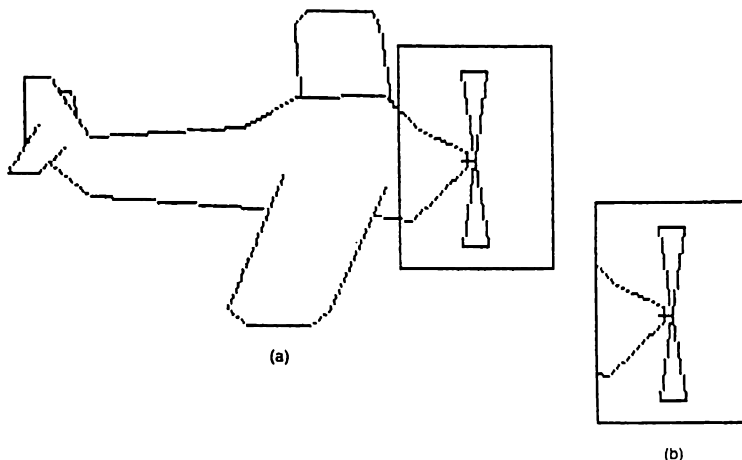
```
2390    '^^^^^^^^^^^^ Program znajdowania przecięć ^^^^^^^^^^^^^^
2400 E1 = E1 + 1
2410 IF X1(P,E+1) <> X1(P,E) THEN 2440
2420 X2(P1,E1) = X1(P,E)      'Linia pionowa
2430 GOTO 2460
2440 M = (Y1(P,E+1) -Y1(P,E)) / (X1(P,E+1) - X1(P,E))
2450 X2(P1,E1) = (B - Y1(P,E)) / M + X1(P,E)
2460 Y2(P1,E1) = B
2470 RETURN
2480    '*************** Narysuj obcięte punkty ***************
2490 CLEARSCREEN
2500 DRAWLINE L,T TO L,B
2510 DRAWLINE L,B TO R,B
2520 DRAWLINE R,B TO R,T
2530 DRAWLINE R,T TO L,T
2540 FOR P = 1 TO MN
2550   FOR E = 1 TO ME(P) - 1
2560     DRAWLINE X2(P,E),Y2(P,E) TO X2(P,E+1),Y2(P,E+1)
2570   NEXT E
2580 NEXT P
2590 RETURN
2600    '*******************************************************
2610 DATA 7
2620    'Zarys
2630 DATA 12,195,127,210,130,232,105,232,100,213,90,200,75,165,77
2640 DATA 145,90,85,95,70,70,60,70,60,97
2650 DATA 3,69,106,85,120,152,125
2660    'Skrzydła
2670 DATA 6,160,110,138,168,145,175,170,175,178,168,200,115
2680 DATA 6,203,80,198,45,195,40,170,40,165,45,167,76
2690    'Ogon
2700 DATA 3,73,75,78,75,80,85
2710 DATA 4,65,90,53,110,65,110,75,100
2720    'Smigło
2730 DATA 7,230,103,235,103,230,65,240,65,230,140,240,140,235,103
2740 END
```

prawego brzegu okna. Do zapamiętania punktów o wartościach współrzędnych X mniejszych niż współrzędna X prawego brzegu okna i punktów przecięć z brzegiem korzysta się powtórnie z tablic X1 i Y1, a do zapamiętania współrzędnych końcowego obrazu wewnątrz okna z tablic X2 i Y2. Na rysunku 8.6 pokazano obraz przed i po obcięciu przez program 8.3.

Rozszerzmy teraz program 8.3 tak, aby mógł obcinać dowolne napisy, które mogą pojawić się na obrazie. Moglibyśmy traktować napisy tak jak linie i zachowywać tę ich część, która znajduje się wewnątrz okna, ale dla uproszczenia obrazu będziemy zachowywać tylko te łańcuchy znaków, które w całości znajdują się we wnętrzu okna. Na rysunku 8.7 zostanie zachowany jedynie NAPIS 3. W celu zachowania łańcucha tekstowego należy sprawdzić, czy tekst zaczyna się z prawej strony lewego brzegu okna i kończy przed prawym brzegiem okna. Tekst musi ponadto być umieszczony w wierszu znajdującym się między dolnym a górnym brzegiem okna.

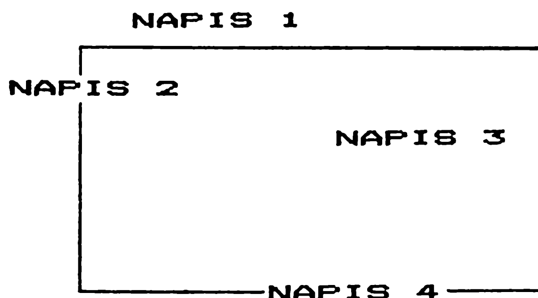
Nasz program używa tablicy T$ do zapamiętania łańcuchów tekstowych. Tablice XL i XR posłużą do zapamiętania współrzędnych X początku i końca każ-

Napisz K aby kontynuować, Z aby zmienić okno



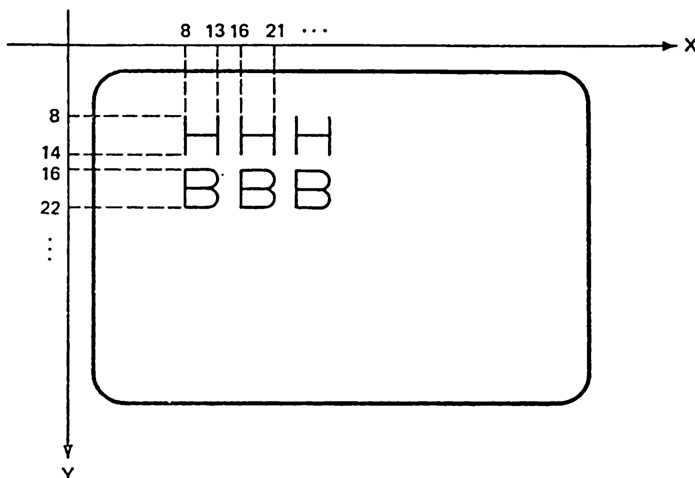
Rys. 8.6. Obraz przed (a) i po (b) obcięciu za pomocą programu 8.3

dego łańcucha. W tablicach YT i YB będą zapamiętywane współrzędne Y góry i dołu łańcuchów. Wartości XL, XR, YT i YB oblicza się na podstawie informacji o wymiarach siatki znakowej dla konkretnego systemu (p. 3.2).

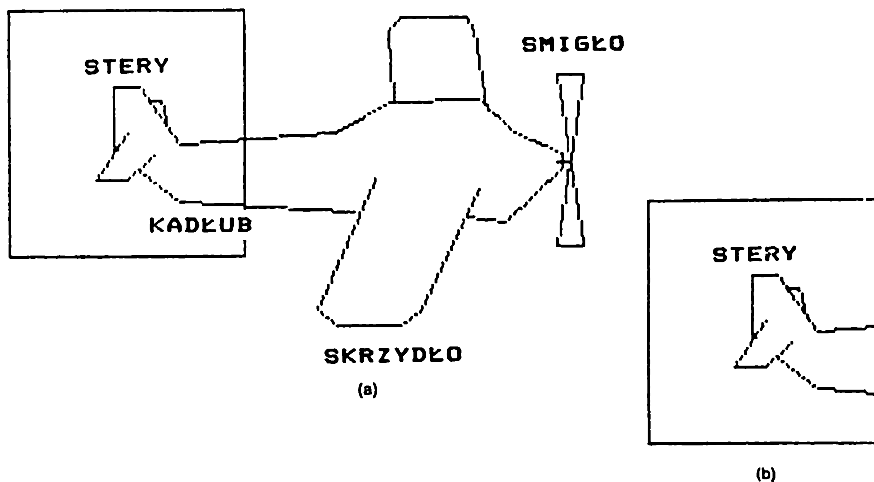


Rys. 8.7. Program obcinania łańcuchów znaków nie leżących w całości wewnątrz okna spowoduje usunięcie napisów: NAPIS 1, NAPIS 2 i NAPIS 4

W naszym programie obcinania tekstu przyjmiemy, że każdy znak mieści się na siatce znakowej o wymiarach 8 × 8 pikseli. Przyjmiemy także, że każdy znak zajmuje obszar 6 × 7 pikseli siatki. Pozostawia to poziomy odstęp 2 pikseli między znakami i pionowy odstęp 1 piksela między wierszami tekstu (rys. 8.8). Znaki na-



Rys. 8.8. Współrzędne znaków umieszczonych w siatce znakowej o wymiarach 8×8 pikseli. W poziomie znaki zaczynają się w punktach 0, 8, 16, ... i kończą w punktach 5, 13, 21, ... Góra znaków zaczyna się w liniach 0, 8, 16, ..., dół znaków w liniach 6, 14, 22 itd.



Rys. 8.9. Rysunek z tekstem przed (a) i po (b) obcinaniu

pisane w poziomym wierszu zaczynają się w punktach 0, 8, 16, 24, ... i kończą w punktach 5, 13, 21 itd. Górne piksele kolejnych wierszy znaków znajdują się w liniach 0, 8, 16, ..., a dolne w liniach 6, 14, 22 itd. W celu określenia współrzędnej X początku znaku odejmujemy 1 od numeru kolumny znaku i mnożymy wynik przez 8. W celu otrzymania końca znaku dodajemy 5 do współrzędnej X początku znaku. Dla znaku napisanego w pozycji 6, 21 współrzędna X początku wynosi 160, a współrzędna X końca 165. W celu określenia współrzędnej Y góry znaku odejmujemy 1 od numeru wiersza znaku i wynik mnożymy przez 8. Dodanie 6 do współrzędnej Y góry znaku daje współrzędną Y dołu znaku. Dla znaku na pozycji 6, 21 współrzędna Y góry znaku wynosi 40, a współrzędna Y dołu znaku 46. Dodatek do programu 8.3 umożliwiający obcinanie tekstów przedstawiono jako program 8.4. Wynik działania programu pokazano na rys. 8.9.

Program 8.4. Obcinanie punktów, linii i napisów na rysunku samolotu

```
10  'PROGRAM 8.4. Obcinanie tekstu i rysunku (uzupełnienie prog. 8.3)
171 DIM T$(10),RO(10),CO(10),TR(10),TL(10),TT(10),TB(10)
271     'Czytaj tekst, numer wiersza i kolumny
272 READ TN              'TN jest liczbą elementów tekstu
273 FOR K = 1 TO TN
274   READ T$(K),RO(K),CO(K)   'RO i CO są nr wiersza i kolumny tekstu
275   TL(K) = (CO(K) - 1) * 8  'Przekształć na pozycję piksela
276   TR(K) = TL(K) - 1 + LEN(T$(K)) * 8
277   TT(K) = (RO(K) - 1) * 8
278   TB(K) = TT(K) + 7
279 NEXT K

315 GOSUB 2461            'Obetnij tekst

431     'Umieść elementy tekstu (część programu rysowania)
432 FOR K = 1 TO TN
433   POSITION RO(K),CO(K)
434   PRINT T$(K)
435 NEXT K

2461 C1 = 0               'C1 jest liczbą elementów tekstu w oknie
2462 FOR K = 1 TO TN
2463   IF TL(K)<L OR TR(K)>R OR TT(K)<T OR TB(K)>B THEN 2448
2464   C1 = C1 + 1
2465   T$(C1) = T$(K)
2466   RO(C1) = RO(K)
2467   CO(C1) = CO(K)
2468 NEXT K
2469 RETURN

2581     'Dodaj elementy tekstu
2582 FOR K = 1 TO C1
2583   POSITION RO(K), CO(K)
2584   PRINT T$(K)
2585 NEXT K

2731     'Elementy tekstu
2732 DATA 4
2733 DATA KADŁUB,14,10
2734 DATA STERY,7,4
2735 DATA SKRZYDŁO,20,20
2736 DATA SMIGŁO,6,28
```

Program 8.4 może być zaadoptowany do innych systemów przez zmianę wymiarów siatki znakowej. Można także zmodyfikować ten program tak, aby zachowywał każdą część łańcucha znaków, która znajduje się w oknie. W celu przyspieszenia wykonywania obliczeń wprowadza się ulepszenie w obcinaniu linii — zwłaszcza jeśli liczba linii do obcinania jest bardzo duża. Na przykład można przetwarzać każdy punkt obrazu tylko raz, zamiast wielokrotnie przy każdej linii. Każdemu punktowi przypisuje się kod oznaczający jego względną pozycję w stosunku do okna. Kodu tego używa się do sprawdzania położenia linii w stosunku do okna. W niektórych systemach graficznych obcinanie odbywa się automatycznie po określeniu położenia i wymiarów okna.

Obraz w oknie po obcięciu może podlegać różnym transformacjom. Możemy na przykład wyświetlić okno w innym położeniu, obrócić je lub zmienić skalę. Możemy także zachować oryginalny obraz i nałożyć okno — być może powiększone — w jednym rogu ekranu. W następnej części omówimy metody transformacji okien.

8.3. Pole prezentacji

Po określeniu położenia i wymiarów okna na obrazie możemy okno przesuwać i zmieniać jego wymiary (skalować) w celu umieszczenia w zadanym prostokątnym obszarze na ekranie. Ten obszar jest zwany *polem prezentacji.* Na ekranie monitora można zdefiniować zarówno okno, jak i pole prezentacji przez podanie współrzędnych górnego lewego rogu i wymiarów (rys. 8.10).

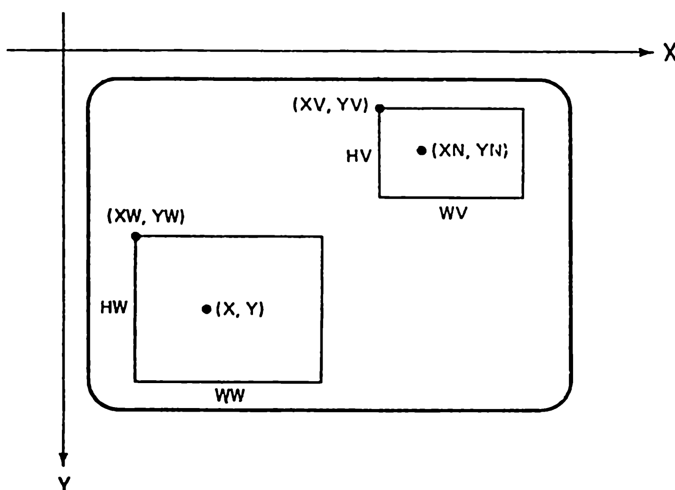
Okno określa, „co" chcemy zobaczyć na obrazie, a pole prezentacji określa, „gdzie" na ekranie chcemy to widzieć. Pole prezentacji ma dowolne wymiary w stosunku do wymiarów okna. Może obejmować cały ekran lub może być małą wstawką w obrazie. Okno i pole prezentacji mogą znajdować się w rozłącznych częściach ekranu lub też mogą się pokrywać. Jeśli pole prezentacji służy do powiększenia części rysunku, to stosuje się je do oglądania małych elementów, które są słabo widoczne w oryginalnym obrazie.

Program transformacji obszaru okna w obszar pola prezentacji musi przekształcić współrzędne (X, Y) każdego punktu w oknie (rys. 8.10) w odpowiadający mu punkt (XN, YN) w polu prezentacji. Transformację tę wykonuje się metodami opisanymi w rozdz. 4, służącymi do wyświetlania całości rysunku na ekranie. Zależności między współrzędnymi punktu w polu prezentacji a współrzędnymi punktu w oknie są następujące:

$$\begin{aligned} XN &= (X - XW) * (WV/WW) + XV \\ YN &= (Y - YW) * (HV/HW) + YV \end{aligned} \tag{8.1}$$

Czynniki (WV/WW) i (HV/HW) są współczynnikami skali. Jeżeli nie są one równe 1,

to następuje zmiana wymiarów rysunku przy przejściu od okna do pola prezentacji. Wartości współczynników skali większe niż 1 odpowiadają powiększeniu okna, wartości mniejsze niż 1 odpowiadają zmniejszeniu okna. Problemy różnego skalowania dla kierunków X i Y omówiono w rozdz. 6. Składniki XV i YV odpowiadają przesunięciu. Jeśli te składniki są różne od XW i YW, to nastąpi przesunięcie fragmentu rysunku.



Rys. 8.10. Określenie okna i pola prezentacji. Górny lewy róg okna znajduje się w punkcie o współrzędnych (XW, YW), a górny lewy róg pola prezentacji w (XV, YV). Wymiary okna i pola prezentacji określa się przez podanie szerokości (WW oraz WV) i wysokości (WH oraz VH). Punkt w oknie o współrzędnych (X, Y) zostanie przeniesiony do nowego położenia (XN, YN) w polu prezentacji

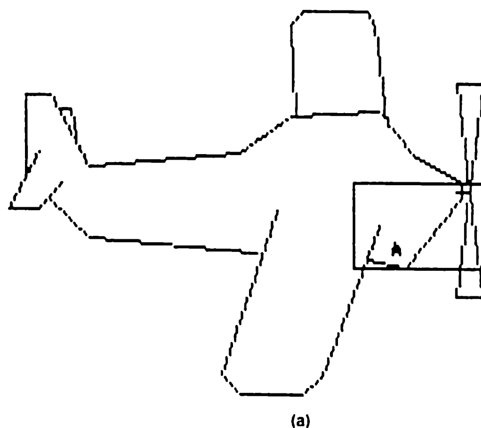
Program 8.5 będący uzupełnieniem programu 8.3 realizuje przeniesienie obszaru okna do zadanego pola prezentacji. Na rysunku 8.11 przedstawiono wynik pracy programu – powiększenie fragmentu rysunku. Gwiazda na kadłubie samolotu widoczna jako niewyraźny zbiór punktów na oryginalnym rysunku jest dobrze widoczna w powiększeniu. Takie powiększenia w polu prezentacji mogą być użyteczne do pokazywania szczegółów na mapach lub złożonych rysunkach.

W niektórych zastosowaniach może wystąpić potrzeba przedstawiania na ekranie jednocześnie oryginalnej sceny i części obrazu w polu prezentacji. Za pomocą okien można wyciąć więcej niż jeden obszar z obrazu i prezentować je w więcej niż jednym polu prezentacji. Kolejna transformacja obszaru wyciętego przez okno może polegać na obracaniu pola prezentacji, co umożliwi przedstawienie wybranego fragmentu w innej orientacji.

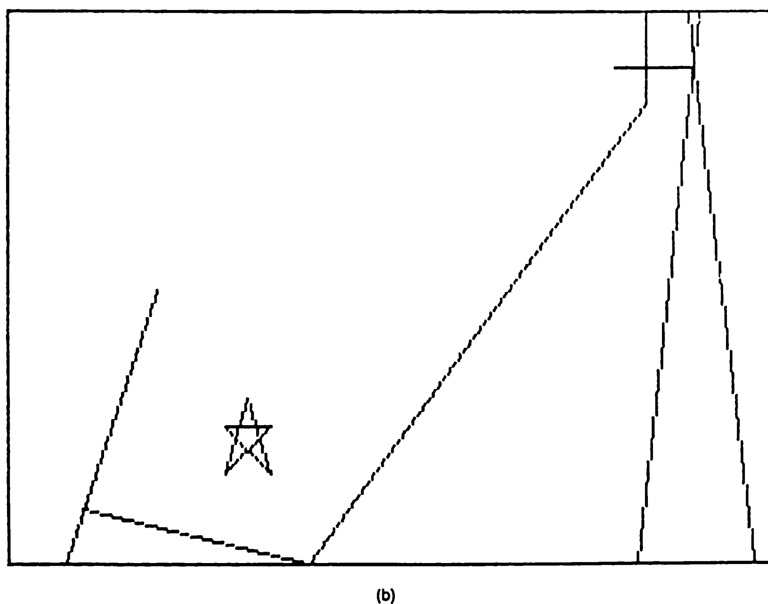
Program 8.5. Rysunek samolotu w polu prezentacji

```
2740 'PROGRAM 8.5. Wyświetlanie obszaru okna w polu prezentacji
2750     '(uzupełnienie do programu 8.3)
2760     'Program ustala pole prezentacji na ekranie. Części obrazu
2770     'wewnątrz okna przenosi do wybranego pola prezentacji
2780     '
2790     '*********************************************************
2800 DIM XN(8,20), Y(8,20)
2810 GOSUB 2850              'Ustal pole prezentacji
2820 GOSUB 3070     'Przenieś obszar okna do obszaru pola prezentacji
2830 GOSUB 3150     'Narysuj części obrazu w polu prezentacji
2840 GOTO 3280
2850    '***************** Ustal pole ******************
2860 POSITION 1,1
2870 PRINT "Górny, lewy róg pola prezentacji ";
2880 INPUT XV, YV
2890 POSITION 1,1
2900 PRINT "                                             ";
2910 POSITION 1,1
2920 PRINT "Szerokość i  wysokość pola prezentacji ";
2930 INPUT WV, HV
2940 POSITIOON 1,1
2950 PRINT "                                             ";
2960 L = XV
2970 R = XV + WV
2980 T = YV
2990 B = YV + HV
3000 IF L<R AND L>=0 AND R<=XM AND T<B AND T>=0 AND B<=YM THEN 3060
3010 POSITION 1,1
3020 PRINT "Pole prozentacji poza ekranem. Spróbuj jeszcze raz"
3030 POSITION 1,1
3040 PRINT "                                             ";
3050 GOTO 2860
3060 RETURN
3070    '******** Przenieś obszar okna do pola prezentacji ***********
3080 FOR P = 1 TO MN
3090   FOR E = 1 TO ME(P)
3100     XN(P,E) = (X2(P,E) - XW) * (WV / WW) + XV
3110     YN(P,E) = (Y2(P,E) - YW) * (HV / HW) + YV
3120   NEXT E
3130 NEXT P
3140 RETURN
3150    '*************** Narysuj obcięte punkty *****************
3160 CLEARSCREEN
3170 DRAWLINE L,T TO L,B
3180 DRAWLINE L,B TO R,B
3190 DRAWLINE R,B TO R,T
3200 DRAWLINE R,T TO L,T
3210 FOR P = 1 TO MN
3220   FOR E = 1 TO ME(P) - 1
3230     DRAWLINE XN(P,E),YN(P,E) TO XN(P,E+1),YN(P,E+1)
3240   NEXT E
3250 NEXT P
3260 RETURN
3270    '**********************************************************
3280 END
```

Napisz K aby kontynuować, Z aby zmienić okno



(a)



(b)

Rys. 8.11. Obszar okna (a) powiększony do wymiarów pola prezentacji (b) za pomocą programu 8.5

ZADANIA

8.1. Napisz program wyróżniania, który zaakcentuje linie wewnątrz wyróżnionego obszaru (okręgu lub prostokąta) rysując je jaśniej (lub w innym kolorze). Dodatkowo dopuść zmianę koloru tła wewnątrz wyróżnionego obszaru.

8.2. Opracuj program wyróżniania obszarów obrazu leżących wewnątrz elipsy o zadanych wymiarach. Napisz opcjonalną procedurę do wymazywania wyróżnionego obszaru.

8.3. Napisz procedurę obcinania łańcucha znaków wewnątrz prostokątnego obszaru.

8.4. Napisz program obcinania, który wymaże wszystkie elementy obrazu poza zadanym obszarem o kształcie okręgu.

8.5. Napisz program obcinania, w którym w polu prezentacji będą przedstawione pewne części obrazu w powiększeniu. Pole prezentacji należy wyświetlić w rogu obrazu, na tle oryginalnego obrazu, wymazując tylko część oryginalnego obrazu zasłoniętą przez pole prezentacji.

8.6. Zmodyfikuj program 8.3 tak, aby każdy punkt obrazu był porównywany z granicą okna tylko raz. Położenie każdego punktu w stosunku do okna może być opisane w tablicy określającej, że punkt leży „Wewnątrz", „na Zewnątrz", „Ponad", „poNiżej", „na Lewo" lub „na pRawo". Informacja o położeniu punktu może być kodowana w postaci łańcucha 3 znaków (na przykład łańcuch „ZNR" będzie oznaczać, że punkt leży na Zewnątrz, poNiżej i na pRawo). Po przetworzeniu wszystkich punktów w celu określenia położenia każdego z nich linie w obrazie mogą być obcinane przez sprawdzanie łańcuchów położenia ich końców. Linia jest zachowywana, jeśli obydwa jej końce są „Wewnątrz". Linia jest eliminowana, jeśli obydwa jej końce są „na Lewo" lub „Ponad" itd. Następnie znajduje się punkty przecięć dla linii nakładających się. Długość łańcucha można zmniejszyć do dwóch znaków, jeśli przyjmuje się, że pierwszy znak może mieć znaczenie „W" (wewnątrz), „P" (powyżej) lub „N" (poniżej), a drugi znak znaczenie „L" (na lewo) lub „R" (na prawo).

8.7. Napisz program, który wyświetli dowolną liczbę fragmentów obrazu na ekranie w różnych polach prezentacji.

8.8. Napisz program obracający pole prezentacji do zadanej orientacji na ekranie.

Część IV

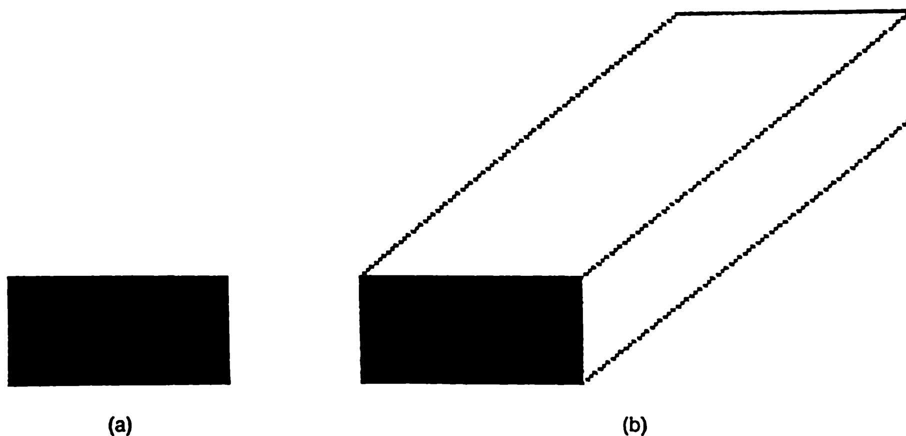
ZAAWANSOWANE TECHNIKI GRAFICZNE

(W trzech wymiarach)

Obiekty rzeczywiste są spostrzegane w trzech wymiarach: oprócz szerokości i wysokości mają także głębię. Przy reprezentacji obiektów na płaskim (dwuwymiarowym) ekranie możemy zignorować głębię lub przedstawić te obiekty w takiej projekcji, aby głębię ukazać. Dodanie głębi znacznie zwiększa realizm sceny lub zawartość informacyjną obrazu. Poniżej rozważymy niektóre metody dodania trzeciego wymiaru w obrazach.

9. Obrazy trójwymiarowe

Trójwymiarowe obiekty, na przykład pudełko, mogą być wyświetlone na ekranie jako prostokąt (rys. 9.1a). Z tej płaskiej reprezentacji otrzymujemy informację jedynie o wysokości i szerokości obiektu; możemy wprawdzie domyślać się, że ma on

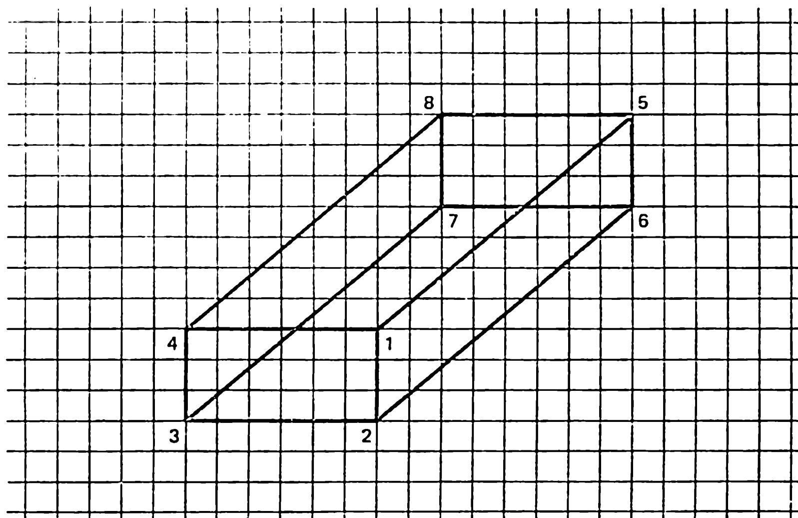


Rys. 9.1. Trójwymiarowe pudełko przedstawione jako płaski prostokąt (a) i z dodaniem głębi (b)

także głębokość, ale oglądamy tylko jeden jego bok. Bardziej realistyczną projekcję pudełka na ekran przedstawiono na rys. 9.1b. Widzimy także inne boki pudełka, możemy więc łatwiej ocenić jego kształt i wymiary. Podobnie, wyświetlanie wykresów i rysunków w trzech wymiarach umożliwia umieszczenie dodatkowych informacji i zależności.

9.1. Rozmieszczanie rysunku na papierze milimetrowym

Obraz pudełka (rys. 9.1) otrzymano przez przeniesienie rysunku z papieru milimetrowego (rys. 9.2) na ekran. Wybieramy taki widok pudełka, aby widoczne były trzy ściany. Z rysunku 9.2 nie wynika jednakże, które ściany są widoczne. Pudełko może być tak widziane z góry z prawej strony lub z dołu z lewej strony. W celu usunięcia tej niejednoznaczności na ekranie pominięto te linie, które znajdują się z tyłu pudełka w wybranym widoku. Narysowaliśmy pudełko bez linii łączących punkty 3 i 7, 6 i 7 oraz 7 i 8. Ta metoda może być też użyta do przedstawiania scen trójwymiarowych: rysujemy wszystkie krawędzie obiektu lub grupy obiektów na papierze milimetrowym, a następnie wymazujemy te linie, które są niewidoczne w wybranym widoku i rysujemy pozostałe linie na ekranie.



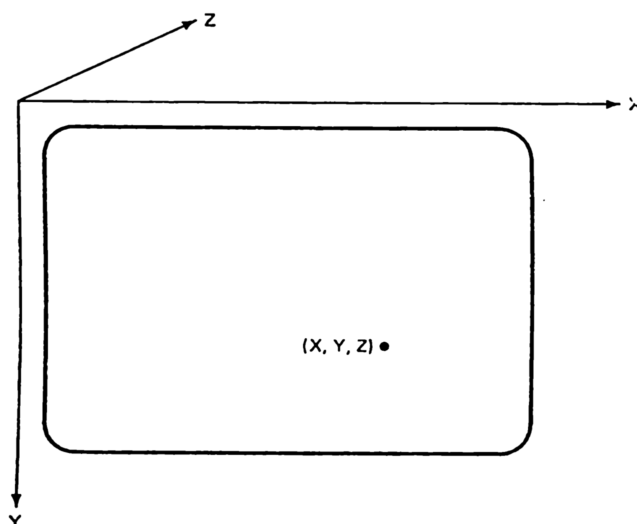
Rys. 9.2. Rysunek obiektu trójwymiarowego na pokratkowanym papierze

W pewnych sytuacjach wolelibyśmy mieć bardziej ogólną metodę otrzymywania widoku obiektów trójwymiarowych, która umożliwiłaby programowi wyświetlanie „przodu" lub „tyłu" obiektów. W bryle obracającej się tak, że kolejne jej ściany pojawiają się w polu widzenia, zmieniają się ciągle ściany przednie na tylne. W celu dostarczenia informacji o głębi obiektów używa się różnych metod w programach wyświetlania. Jedna z nich polega na usunięciu linii niewidocznych, tak jak to zrobiliśmy na papierze. Możemy także przedstawić obiekty w rzucie perspektywicznym w celu przekazania informacji o ich głębi. W rzucie perspektywicznym części obiektów położone bliżej nas są widziane jako większe, a części położone dalej

jako mniejsze. Inna technika polega na zwiększaniu jasności linii bliżej położonych. Złożone metody cieniowania umożliwiają łagodne przejście od świateł do cieni na ścianach obiektów. Każda z tych metod wymaga wprowadzania do programu wyświetlania informacji o głębokości punktów obrazu.

9.2. Współrzędne w trzech wymiarach

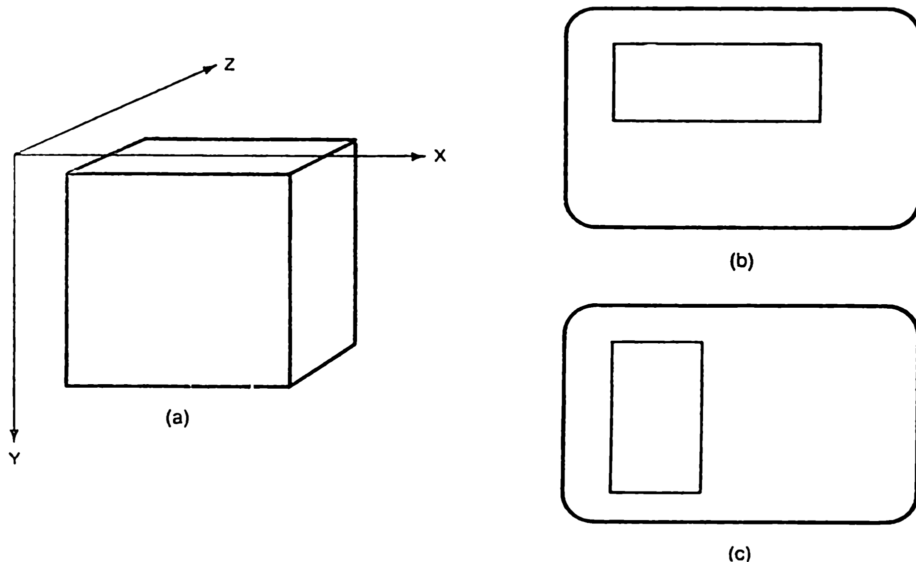
Na rysunku 9.3 pokazano przyjęty sposób podawania współrzędnych trójwymiarowych, w którym współrzędna Z odpowiada głębokości. Jako początek układu współrzędnych wybrano lewy górny róg ekranu. Tak jak poprzednio, wartości współrzęd-



Rys. 9.3. Każdy punkt obrazu wymaga podania trzech wartości współrzędnych (X, Y, Z). Dodatnie wartości Z oznaczają przestrzeń w głąb ekranu

nej X rosną od lewej do prawej strony ekranu, a współrzędnej Y od góry w dół. Współrzędna Z wzrasta w głąb – wartości dodatnie odpowiadają przestrzeni poza ekranem, a ujemne przestrzeni przed ekranem. Każdy punkt na rysunku jest określony przez trzy współrzędne (X, Y, Z). O położeniu punktu na ekranie decydują wartości X i Y, a głębokość położenia punktu w stosunku do ekranu jest wyznaczona przez wartość Z. Punkty położone dalej mają wartości Z większe. a punkty położone bliżej wartości mniejsze. Współrzędna Z jest używana do otrzymania różnych widoków obiektów, do identyfikacji linii, powierzchni lub obiektów niewidocznych oraz do przedstawiania obiektów w rzucie perspektywicznym.

Różne widoki obiektów otrzymujemy podstawiając współrzędną Z każdego punktu zamiast współrzędnej X lub Y w trakcie rysowania na ekranie. Rysując współrzędne (X, Z) wszystkich punktów otrzymujemy widok obiektu z góry lub z dołu, a rysując współrzędne (Z, Y) otrzymujemy widok z boku (rys. 9.4). W pierwszym przypadku (widok z góry) przyjmujemy, że wartości Y są dodatnie z tyłu ekranu; w przeciwnym razie otrzymujemy widok z dołu. W drugim przypadku otrzymujemy widok z lewej strony, jeśli przyjmiemy, że wartości X są dodatnie z tyłu ekranu, i widok z prawej strony, jeśli przyjmiemy, że wartości X są dodatnie z przodu ekranu. Takie widoki są zwane *rzutami prostokątnymi* (projekcjami). W celu otrzymania widoków obiektów z góry, z dołu lub z boku wprowadzamy współrzędne (X, Y, Z) każdego wierzchołka i wykreślamy obiekt według pary współrzędnych, eliminując „tylne" płaszczyzny.



Rys. 9.4. Rzuty prostokątne: obiekt trójwymiarowy (a) widziany z góry lub z dołu przez wykreślenie jedynie wartości współrzędnych (X, Z) na ekranie (b); widok boczny (c) otrzymuje się przez wykreślenie wartości współrzędnych (Z, Y)

9.3. Wymazywanie niewidocznych powierzchni i linii

Istnieją dwie ogólne metody wymazywania niewidocznych części obiektów. Jedna z metod polega na wyobrażeniu obiektu jako zbioru powierzchni. Możemy wtedy określić i usunąć te powierzchnie, które znajdują się z tyłu, ukryte za innymi powierzchniami. Druga metoda polega na traktowaniu obrazu jako zbioru odcinków i określeniu oraz wymazaniu niewidocznych odcinków — nie powierzchni.

Niewidoczne powierzchnie

Technika eliminacji powierzchni niewidocznych na obrazie polega na rysowaniu na ekranie wszystkich powierzchni, poczynając od tylnych a kończąc na przednich. Powierzchnie o większych wartościach Z są rysowane najpierw, tak że powierzchnie bliższe zasłaniają te tylne, jeśli na nie nachodzą. Każda powierzchnia może być rysowana w innym kolorze lub wszystkie mogą być rysowane w kolorze tła, co powoduje usunięcie niewidocznych linii.

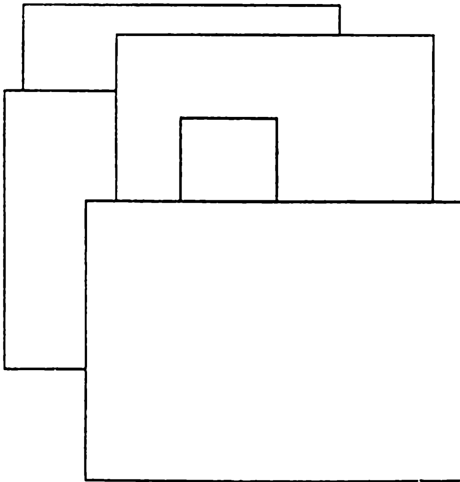
Program 9.1 stanowi ilustrację metody wymazywania niewidocznych linii. Do programu wprowadza się dowolną liczbę prostokątnych ścian. Przyjmujemy, że każda ściana płaska jest równoległa do płaszczyzny ekranu, tak że dla każdej ściany podaje się tylko jedną wartość Z. Po narysowaniu kolejnej ściany — poczynając od najdalszej, a na najbliższej kończąc — jej wnętrze jest wypełniane kolorem tła.

Program 9.1. Wymazywanie niewidocznych linii metodą kreślenia ścian na ekranie poczynając od położonych najdalej (ściany prostokątne)

```
10  'PROGRAM 9.1. Wymazywanie ścian przez wypełnianie ich kolorem tła
20      'Program zapamiętuje współrzędne X, Y i Z górnego lewego rogu
30      'pewnej liczby powierzchni prostokątnych oraz ich szerokość
40      'i wysokość. Następnie sortuje te powierzchnie w porządku
50      'malejących wartości Z i rysuje je. Prostokąty są wypełniane
60      'kolorem tła, co powoduje, że każdy prostokąt wymazuje
70      'tę część innych prostokątów, które są za nim.
80      '**************************************************************
90  DIM X(10), Y(10), Z(10), W(10), H(10)
100 XM = 255
110 YM = 191        'XM i YM są maksymalnymi wartościami dla systemu
120     '************* Odczytaj dane prostokątów ****************
130 READ N          'N jest liczbą prostokątów
140 FOR K = 1 TO N
150   READ X(K), Y(K), Z(K), W(K), H(K)
155     'X, Y - współrzędne lewego rogu, H - szerokość, W - wysokość
160   IF X(K)<0 OR Y(K)<0 OR W(K)<0 OR H(K)<0 THEN 650
170   IF X(K)+W>XM OR Y(K)+H>YM THEN 650
180 NEXT K
190     '****** Posortuj prostokąty wg wartości współrzędnej Z *******
200 FOR P = 1 TO N -1
210   B = P
220   FOR R = P+ 1 TO N             'Przejrzyj pozostałe wartości Z
230     IF Z(B) > Z(R) THEN 250     'Jeśli mniejsze - kontynuuj
240     B = R                       'Jeśli nie  - podstaw większą
250   NEXT R
260   T = X(P)
270   X(P) = X(B)
280   X(B) = T
290   T = Y(P)
300   Y(P) = Y(B)
310   Y(B) = T
320   T = Z(P)
330   Z(P) = Z(B)
340   Z(B) = T
350   T = W(P)
360   W(P) = W(B)
370   W(B) = T
380   T = H(P)
390   H(P) = H(B)
```

Program 9.1 (cd.)

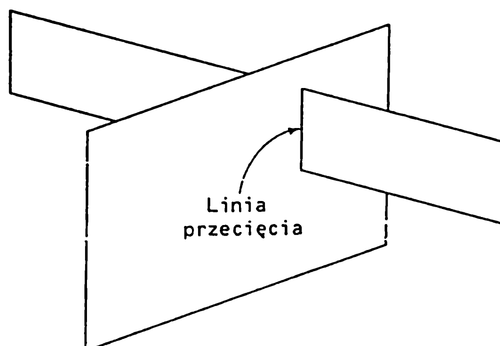
```
400    H(B) = T
410 NEXT P
420    '******* Narysuj prostokąty w kolejności posortowania *******
430 CLEARSCREEN
440 GRAPHICS
450 FOR K = 1 TO N
460    DRAWLINE X(K),Y(K) TO X(K)+W(K),Y(K)
470    DRAWLINE X(K)+W(K),Y(K) TO X(K)+W(K),Y(K)+H(K)
480    DRAWLINE X(K)+W(K),Y(K)+H(K) TO X(K),Y(K)+H(K)
490    DRAWLINE X(K),Y(K)+H(K) TO X(K),Y(K)
500      'Wypełnij narysowany obszar
510    COLOR 0,0
520    FOR YI = Y(K) + 1 TO Y(K) + H(K) - 1
530      DRAWLINE X(K)+1,YI TO X(K)+W(K)-1,YI
540    NEXT YI
550    COLOR 1,0
560 NEXT K
570      '*************************************************************
580 DATA 5
590 DATA 10,10,30,100,100
600 DATA 40,20,2,100,100
610 DATA 60,50,0,30,100
620 DATA 5,40,20,50,100
630 DATA 30,80,-5,120,100
640      '*************************************************************
650 END
```



Rys. 9.5. Nakładające się ściany wyświetlone przez program 9.1

Ściany zachodzące na inne wymażą te dalsze (rys. 9.5). Użycie różnych kolorów dla różnych ścian umożliwia zamalowanie niewidocznych linii, zamiast ich wymazywania.

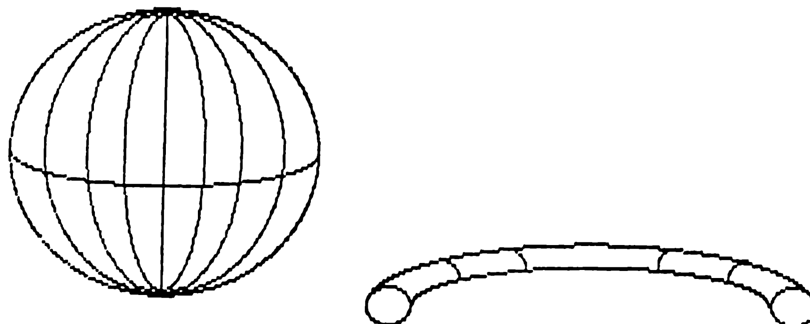
Do programu można wprowadzić także inne kształty ścian np. okręgi, trójkąty lub dowolne wielokąty. Długość linii brzegowej, która zostanie wymazana, zależy od równań brzegów tych ścian. Ponadto, zamiast podawać wspólną współrzędną Z dla całej ściany, można ją podać oddzielnie dla każdego jej punktu. Umo-



Rys. 9.6. Płaszczyzny o zmiennej współrzędnej Z mogą pokrywać się i przecinać tak, że wzajemnie się zasłaniają

żliwia to przedstawienie ścian nachylonych, z których pewne części są położone dalej od nas niż inne. Następnie konieczne będzie wykonanie sortowania wartości Z każdej ściany w celu znalezienia najmniejszej wartości. Możemy także żądać, aby ściany nie przecinały się. W przeciwnym przypadku dwie ściany mogą się nawzajem zasłaniać (rys. 9.6). W takim przypadku trzeba znaleźć linię przecięcia ścian i podzielić jedną z nich lub obydwie na dwie części albo zastosować inne metody geometrii w celu ustalenia, która część każdej ściany jest widoczna. W końcu, aby określić wzajemne zależności między ścianami krzywoliniowymi należałoby je podzielić na pewną liczbę składowych ścian płaskich. W celu wskazania krzywizny ściany można jej całe wnętrze wymazać i narysować dodatkowe linie wewnętrzne (rys. 9.7).

Jako alternatywę wymazywania wnętrza wszystkich ścian w obrazie można stosować metodę identyfikacji i eliminacji tylko tych powierzchni, które są rzeczywiście niewidoczne. W przypadku obiektów symetrycznych zwykle udaje się znaleźć metody umożliwiające określenie widoczności ścian przeciwległych. Pudełko z rys. 9.2 ma trzy pary ścian przeciwległych. W danej chwili możemy widzieć tylko jedną ścianę każdej z tych par. Jeśli widzimy ścianę o wierzchołkach 1, 2, 3 i 4, to nie możemy



Rys. 9.7. Przedstawienie trójwymiarowych powierzchni krzywoliniowych za pomocą dodatkowych linii uwydatniających krzywizny

Program 9.2. Eliminacja niewidocznych linii przez wyświetlanie tylko ściany widocznej z każdej pary symetrycznych ścian obiektu (pudełka)

```
10  'PROGRAM 9.2. Wymazywanie linii niewidocznych w obiekcie o symetry-
20       'cznych ścianach. Program czyta i zapamiętuje wierzchołki trój-
30       'wymiarowego pudełka. Następnie porównuje wartości Z dla każdej
40       'pary symetrycznych ścian i rysuje ścianę położoną bliżej.
50  CLEARSCREEN
60  DIM X(8), Y)(8, Z(8)
70  XM = 39
80  YM = 39            'XM i YM są maksymalnymi wartościami dla systemu
90       '************* Odczytaj wierzchołki pudełka ***************
100 FOR K =1 TO 8
110   READ X(K), Y(K), Z(K)
120   IF X(K) < 0 OR X(K) > XM OR Y(K) < 0 OR Y(K) > YM THEN 650
130 NEXT K
140      '************** Narysuj wszystkie ściany ***************
150 GRAPHICS
160 FOR K = 1 TO 3
170   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
180   DRAWLINE X(K),Y(K) TO X(K+4),Y(K+4)
190 NEXT K
200 DRAWLINE X(4),Y(4) TO X(1),Y(1)
210 DRAWLINE X(4),Y(4) TO X(8),Y(8)
220 FOR K = 5 TO 7
230   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
240 NEXT K
250 DRAWLINE X(8),Y(8) TO X(5),Y(5)
260 FOR K = 1 TO 1000: NEXT K
270      '************** Narysuj tylko ściany widoczne ************
280 CLEARSCREEN
290 IF Z(1) = Z(5) THEN 400          'Żadna ściana nie jest widoczna
300 IF Z(1) > Z(5) THEN 360
310 FOR K = 1 TO 3                   'Narysuj ścianę zawierającą punkt 1
320   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
330 NEXT K
340 DRAWLINE X(4),Y(4) TO X(1),Y(1)
350 GOTO 410
360 FOR K = 5 TO 7                       'Narysuj ścianę zaw. punkt 5
370   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
380 NEXT K
390 DRAWLINE X(8),Y(8) TO X(5),Y(5)
400 IF Z(1) = Z(4) THEN 510          'Żadna ściana nie jest widoczna
410 IF Z(1) > Z(4) THEN 470
420 DRAWLINE X(1),Y(1) TO X(2),Y(2)      'Narysuj ścianę zaw. punkt 1
430 DRAWLINE X(2),Y(2) TO X(6),Y(6)
440 DRAWLINE X(6),Y(6) TO X(5),Y(5)
450 DRAWLINE X(5),Y(5) TO X(1),Y(1)
460 GOTO 520
470 DRAWLINE X(4),Y(4) TO X(3),Y(3)      'Narysuj ścianę zaw. punkt 4
480 DRAWLINE X(3),Y(3) TO X(7),Y(7)
490 DRAWLINE X(7),Y(7) TO X(8),Y(8)
500 DRAWLINE X(8),Y(8) TO X(4),Y(4)
510 IF Z(1) = Z(2) THEN 650          'Żadna ściana nie jest widoczna
520 IF Z(1) > Z(2) THEN 580
530 DRAWLINE X(1),Y(1) TO X(4),Y(4)      'Narysuj ścianę zaw. punkt 1
540 DRAWLINE X(4),Y(4) TO X(8),Y(8)
550 DRAWLINE X(8),Y(8) TO X(5),Y(5)
560 DRAWLINE X(5),Y(5) TO X(1),Y(1)
570 GOTO 650
580 DRAWLINE X(2),Y(2) TO X(3),Y(3)      'Narysuj ścianę zaw. punkt 2
590 DRAWLINE X(3),Y(3) TO X()7,Y(7)
600 DRAWLINE X(7),Y(7) TO X(6),Y(6)
610 DRAWLINE X(6),Y(6) TO X(2),Y(2)
620      '*********************************************************
630 DATA 19,22,0,19,28,10,11,28,60,11,22,50
640 DATA 29,10,100,29,16,110,21,16,160,21,10,150
650 END
```

widzieć ściany o wierzchołkach 5, 6, 7, 8. Program 9.2 eliminuje niewidoczne linie pudełka wyświetlając tylko tę ścianę z pary, która ma mniejszą wartość Z. Wyświetla on najpierw wszystkie krawędzie testowego pudełka, potem wymazuje ekran i rysuje tylko widoczne ściany. Gdyby pudełko się obracało, program 9.2 musiałby od nowa ustalać widoczność ścian po każdym obrocie. Podobne metody mogą być stosowane do innych obracających się obiektów.

Metodę eliminacji niewidocznych powierzchni, w której nie korzysta się z symetrii obiektu, podano w programie 9.3. Dla każdej ściany obiektu określa się opisany na niej obszar prostokątny. Dla każdego wierzchołka każdej ściany sprawdza się, czy znajduje się on wewnątrz prostokątnego otoczenia każdej z pozostałych ścian. Jeśli wierzchołek leży wewnątrz tego otoczenia i ma większą głębokość, to wskaźnik widoczności ściany zawierającej ten wierzchołek ustawia się na OFF (niewidoczna). Po sprawdzeniu widoczności wszystkich ścian są rysowane jedynie widoczne ściany obiektu. Na rysunku 9.8 pokazano dwa widoki obiektu uzyskane za pomocą programu 9.3.

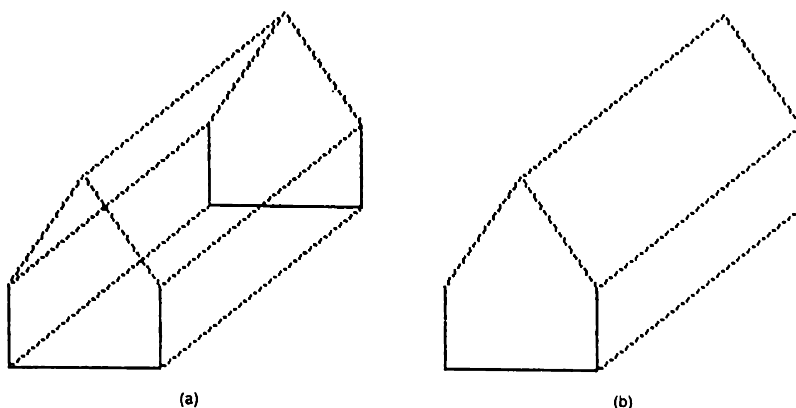
Program 9.3. Usuwanie niewidocznych powierzchni metodą znajdowania zasłoniętych wierzchołków

```
10  'PROGRAM 9.3. Wymazywanie linii niewidocznych za pomocą otoczenia
15                prostokątnego
20     'Wszystkie ściany są początkowo uznane za widoczne (ON) i zostają
30     'narysowane. Znajdujemy największe i najmniejsze wartości X, Y
40     'i Z dla każdej ściany i zapamiętujemy je. Wartości X i Y wyzna-
50     'czają "prostokąt otoczenia" ściany. Biorąc po kolei wszystkie
60     'ściany bada się widoczność innych w stosunku do niej. Jeśli
70     'jakiś wierzchołek badanej ściany znajduje się wewnątrz "prosto-
80     'kąta otoczenia", to porównuje się wartości Z. Jeśli wartość Z
90     'wierzchołka badanej ściany jest większa niż największa wartość
100    'Z ściany testowej, to ścianę testową uznajemy za niewidoczną
110    '(OFF). Jeżeli już wszystkie ściany zostały przetestowane
120    'w stosunku do wszystkich pozostałych, to rysuje się rysunek
130    'złożony tylko z tych ścian, które są ciągle widoczne (ON).
140    '*************************************************************
150 DIM X(9,6), Y(9,6), Z(9,6), C$(9)
160 DIM XS(9), YS(9), ZS(9), XL(9), YL(9), ZL(9)
170 XM = 279
180 YM = 159        'XM, YM są maksymalnymi wartościami dla systemu
190       '
200       '***************** Odczytaj punkty *************************
210 READ N                       'N jest liczbą ścian
220 FOR S = 1 TO N
230   C$(S) = "ON"
240   READ NV(S)                 'NV jest liczbą wierzchołków ściany
250   FOR V = 1 TO NV(S)
260     READ X(S,V), Y(S,V), Z(S,V)
270     IF X(S,V)<0 OR X(S,V)>XM OR Y(S,V)<0 OR Y(S,V)>YM THEN 880
280   NEXT V
290   X(S,NV(S)+1) = X(S,1)
300   Y(S,NV(S)+1) = Y(S,1)
310 NEXT S
320 GRAPHICS
330 GOSUB 680                     'Narysuj widoczne ściany
340       '
350       '*********** Znajdź granice każdej ściany *****************
360 FOR S = 1 TO N
370   XS(S) = X(S,1)             'XS -  wartość najmniejsza
```

Program 9.3 (cd.)

```
380    YS(S) = Y(S,1)              'XL - wartość największa
390    ZS(S) = Z(S,1)
400    FOR V = 2 TO NV(S)
410      IF X(S,V) < XS(S) THEN XS(S) = X(S,V)
420      IF Y(S,V) < YS(S) THEN YS(S) = Y(S,V)
430      IF Z(S,V) < ZS(S) THEN ZS(S) = Z(S,V)
440      IF X(S,V) > XL(S) THEN XL(S) = X(S,V)
450      IF Y(S,V) > YL(S) THEN YL(S) = Y(S,V)
460      IF Z(S,V) > ZL(S) THEN ZL(S) = Z(S,V)
470    NEXT V
480 NEXT S
490      '
500      '************ Niewidoczne ściany oznacz OFF **************
510 FOR S = 1 TO N
520    IF C$(S) = "OFF" THEN 640     'Sciana jest niewidoczna (OFF)
530    FOR R = 1 TO N
540      IF C$(R) = "OFF" OR R = S THEN 630    'Niepotrzebne porównanie
550      FOR V = 1 TO NV(R)
560        IF X(R,V) <= XS(S) OR X(R,V) >= XL(S) OR
              Y(R,V) <= YS(S) OR Y(R,V) >= YL(S) THEN 620
570        'Punkt jest wewnątrz prostokąta otoczenia,
580        'ale z przodu czy z tyłu badanej ściany?
590        IF Z(R,V) <= ZL(S) THEN 630   'Sciana R jest z przodu ściany S
600        C$(S) = "OFF"                 'Sciana R jest z tyłu ściany S
610        GOTO 630                      'Przetestuj następną ścianę
620      NEXT V
630    NEXT R
640 NEXT S
650 GOSUB 680                   'Narysuj ściany widoczne (ON)
660 GOTO 880
670      '
680      '******************** Program rysowania **********************
690 CLEARSCREEN
700 FOR S = 1 TO N
710    IF C$(S) = "OFF" THEN 750         'Pomiń tę ścianę - nie rysuj jej
720    FOR V = 1 TO NV(S)
730      DRAWLINE X(S,V),Y(S,V) TO X(S,V+1),Y(S,V+1)
740    NEXT V
750 NEXT S
760 RETURN
770      '
780      '**********************************************************
790 DATA 7
800 DATA 5,90,140,60,150,140,10,150,110,0,120,70,20,90,110,50
810 DATA 4,150,110,10,150,140,10,230,80,110,230,50,100
820 DATA 5,230,50,100,230,80,110,170,80,160,170,50,150,200,10,120
830 DATA 4,170,50,150,170,80,160,90,140,60,90,110,50
840 DATA 4,90,140,50,150,140,10,230,80,110,170,80,160
850 DATA 4,150,110,0,230,50,100,200,10,120,120,70,20
860 DATA 4,90,110,50,120,70,70,200,10,120,170,50,150
870      '**********************************************************
880 END
```

Obiekty o skomplikowanych kształtach mogą być narysowane niedokładnie za pomocą metod stosowanych w programie 9.3, gdyż test widoczności jest zbyt uproszczony. Metoda dopuszcza jedynie całkowitą widoczność lub całkowite ukrycie każdej ściany, co uniemożliwia wyświetlenie ścian częściowo widocznych. Ponadto, prostokątne otoczenie ściany wykorzystane do testowania widoczności wierzchołków



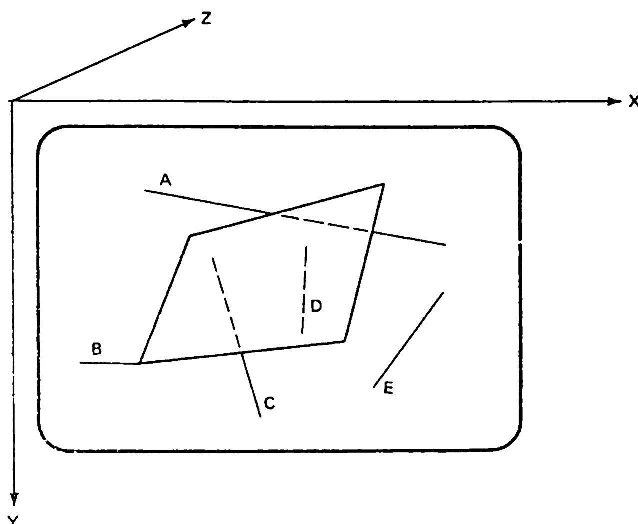
Rys. 9.8. Obiekt trójwymiarowy przed (a) i po usunięciu (b) niewidocznych powierzchni narysowany za pomocą programu 9.3

jest przyczyną tym większej niedokładności, im kształt ściany bardziej odbiega od prostokąta. Długa, wąska, ukośna ściana będzie miała duże otoczenie prostokątne, co może prowadzić do błędnego określenia widoczności.

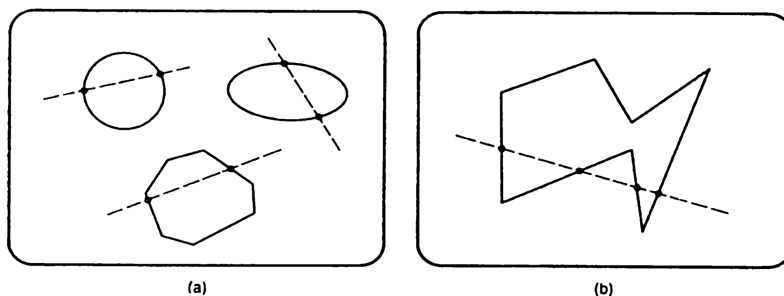
Niewidoczne linie

Dwa poprzednie programy testowały powierzchnie obiektów w celu sprawdzenia ich widoczności. Przyjmowało się, że każda badana powierzchnia jest widoczna lub niewidoczna. Rozważymy teraz metodę określania częściowej widoczności przez badanie widoczności pojedynczych linii zamiast całych powierzchni. Nasz program będzie badał linię w celu określenia, czy jakakolwiek jej część jest ukryta za jakąś powierzchnią.

Linia i powierzchnia mogą być różnie usytuowane w stosunku do siebie. Na rysunku 9.9 przedstawiono różne zależności między położeniem linii i powierzchni. Przy wykonywaniu tego rysunku przyjęto, że powierzchnia jest położona bliżej nas niż linie. Linie B i E są całkowicie widoczne, linia D jest całkowicie niewidoczna, górny odcinek linii C jest też niewidoczny, podobnie jak środkowa część linii A. W celu uproszczenia naszego programu eliminacji niewidocznych linii przyjmiemy, że są to jedyne możliwe relacje. Oznacza to, że linia nie może przecinać powierzchni w więcej niż dwóch punktach. Ogranicza to zbiór kształtów powierzchni do okręgów, elips lub wielokątów wypukłych (rys. 9.10a) i eliminuje z rozważań wszystkie wielokąty wklęsłe (rys. 9.10b), które mogą mieć więcej niż dwa punkty przecięcia z linią prostą. Możemy dopuścić obiekty z wklęsłymi powierzchniami przez ponowne zdefiniowanie granic powierzchni. Dowolna powierzchnia wklęsła



Rys. 9.9. Możliwe relacje między położeniem linii a powierzchnią: linia A przecina granice ściany w dwóch punktach, a linie B i C w jednym; linia D jest całkowicie niewidoczna, a linia E całkowicie widoczna (Przyjęto, że współrzędne Z dla linii są większe niż dla powierzchni)



Rys. 9.10. Okręgi, elipsy i wielokąty wypukłe (kąty wewnętrzne mniejsze niż 180°) mają nie w ęcej niż dwa punkty przecięcia z linią prostą (a); wielokąty wklęsłe (kąty wewnętrzne większe niż 180°) mogą mieć więcej niż dwa punkty przecięcia z linią prostą (b)

może być przedstawiona przy użyciu dwóch lub większej liczby wielokątów wypukłych. Powoduje to jednak dodanie linii do obiektów, co może być niepożądane.

W programie 9.4 przedstawiono metodę wykrywania i wymazywania niewidocznych fragmentów linii. Do programu można wprowadzić dowolną liczbę powierzchni przez podanie współrzędnych ich wierzchołków. W przykładzie użyto

Program 9.4. Usuwanie niewidocznych fragmentów częściowo widocznych linii i powierzchni

```
10  'PROGRAM 9.4. Usuwanie fragmentów linii ścian częściowo widocznych
20    'Końce boków każdej ściany S (nie może być wklęsła) są zapamię-
30    'tywane w tablicach X1, Y1, Z1, X2, Y2, Z2. Program rysuje figurę
40    'i zaczyna usuwać linie niewidoczne od jednej ściany S poczynając.
50    'Znajduje się wartości minimalne i maksymalne X, Y i Z dla okreś-
60    'lenia granic tej ściany. Znajduje się nachylenie i przesunięcie
70    'dla każdej linii tej ściany. Program następnie sprawdza położenie
80    'każdej linii z pozostałych ścian w stosunku do testowanej
90    'ściany. Jeśli linia testowana jest poza granicami ściany, to jest
100   'ona widoczna i przechodzi się do następnej linii. Jeśli nie, to
110   'próbuje się znaleźć punkty przecięcia tej linii i boków ściany S.
120   'Następnie trzeba sprawdzić, czy obliczone punkty przecięcia
130   'rzeczywiście należą do obydwu linii (może być tak, że linia
140   'testowana lub bok ściany S kończą się przed wyliczonym
150   'punktem przecięcia). Następnie porównuje się wartości Z, aby
160   'określić czy testowana linia znajduje się z przodu czy z tyłu
170   'ściany S w obliczonym punkcie przecięcia.  Jeśli znajdziemy
180   'dwa punkty przecięcia ograniczające niewidoczny odcinek linii,
190   'to jest on usuwany. Po sprawdzeniu wszystkich linii figury
200   'w stosunku do wybranej ściany S i usunięciu niewidocznych
210   'odcinków przechodzimy do następnej ściany figury.
220   '
230   ,*****************************************************************
240 CLEARSCREEN
250 DIM X1(14,4), X2(14,4), Y1(14,4), Y2(14,4), Z1(14,4), Z2(14,4)
260 DIM NV(14), M(4), B(4)
270 XM = 319
280 YM = 199
290   '************* Odczytaj współrzędne punktów **************
300 READ NS                      'NS jest liczbą ścian
310 FOR S = 1 TO NS
320   READ NV(S)                 'NV(S) jest liczbą wierzchołków ściany
330   READ X1(S,1), Y1(S,1), Z1(S,1)
340   FOR V = 2 TO NV(S)
350     READ X1(S,V), Y1(S,V), Z1(S,V)
360     IF X1(S,V) < 0 OR X1(S,V) > XM THEN 2300
370     IF Y1(S,V) < 0 OR Y1(S,V) > YM THEN 2300
380     X2(S,V-1) = X1(S,V)
390     Y2(S,V-1) = Y1(S,V)
400     Z2(S,V-1) = Z1(S,V)
410   NEXT V
420   X2(S,NV(S)) = X1(S,1)
430   Y2(S,NV(S)) = Y1(S,1)
440   Z2(S,NV(S)) = Z1(S,1)
450 NEXT S
460   'Narysuj wielokąt ściany
470 GRAPHICS
480 FOR S = 1 TO NS
490   FOR V = 1 TO NV(S)
500     DRAWLINE X1(S,V),Y1(S,V) TO X2(S,V),Y2(S,V)
510   NEXT V
520 NEXT S
530 COLOR 0,0
540 FOR S = 1 TO NS
550   GOSUB 650                  'Znajdź granice i równania ścian
560   FOR SR = 1 TO NS
570     IF S = SR THEN 610
580     FOR L = 1 TO NV(SR)
590       GOSUB 1000         'Sprawdź widoczność, usuń niewidoczne części
600     NEXT L
```

Program 9.4 (cd.)

```
610   NEXT SR
620 NEXT S
630 GOTO 2300
640   '
650   '***** Znajdź granice każdej ściany i równania jej boków *******
660   'Znajdź współrzędne X, Y i Z granic ściany
670 XL = X1(S,1)          'XL jest minimalną wartością współrzędnej X,
680 XR = X1(S,1)          'XR jest maksymalną wartością współrzędnej X
690 YT = Y1(S,1)          'YT jest minimalną wartością współrzędnej Y
700 YB = Y1(S,1)          'YB jest maksymalną wartością współrzędnej Y
710 ZF = Z1(S,1)          'ZF jest minimalną wartością współrzędnej Z
720 FOR K = 1 TO NV(S)
730   IF Z1(S,K) < ZF THEN ZF = Z1(S,K)
740   IF Z2(S,K) < ZF THEN ZF = Z2(S,K)
750   IF Y1(S,K) < YT THEN YT = Y1(S,K)
760   IF Y2(S,K) < YT THEN YT = Y2(S,K)
770   IF Y1(S,K) > YB THEN YB = Y1(S,K)
780   IF Y2(S,K) > YB THEN YB = Y2(S,K)
790   IF X1(S,K) > XR THEN XR = X1(S,K)
800   IF X2(S,K) > XR THEN XR = X2(S,K)
810   IF X1(S,K) < XL THEN XL = X1(S,K)
820   IF X2(S,K) < XL THEN XL = X2(S,K)
830 NEXT K
840   '***************** Określ równania linii ************************
850   'Dla każdej linii figury znajdź nachylenie i przesunięcie.
860   'Zapamiętaj je w tablicach M i B.
870 FOR P = 1 TO NV(S)
880   IF Y1(S,P) = Y2(S,P) THEN 930    'Linia pozioma
890   IF X1(S,P) = X2(S,P) THEN 960    'Linia pionowa
900   M(P) = (Y2(S,P) - Y1(S,P)) / (X2(S,P) - X1(S,P))
910   B(P) = Y1(S,P) - M (P)* X1(S,P)
920   GOTO 970
930   M(P) = 0        'Nachylenie linii poziomej wynosi 0
940   B(P) = Y1(S,P)
950   GOTO 970
960   M(P) = 9999     'Nachylenie linii pionowej nie jest zdefiniowane
970 NEXT P
980 RETURN
990   '
1000  '************** Zbadaj widoczność linii *********************
1010  'Jeśli punkty linii testowej L ściany SR są poza  granicami ściany
1020  'S, to linia jest widoczna w stosunku do ściany S.
1030  '
1040 IF Z1(SR,L)<=ZF AND Z2(SR,L)<=ZF THEN 2130   'Linia jest widoczna
1050 IF Y1(SR,L)<=YT AND Y2(SR,L)<=YT THEN 2130   'Linia jest widoczna
1060 IF Y1(SR,L)>=YB AND Y2(SR,L)>=YB THEN 2130   'Linia jest widoczna
1070 IF X1(SR,L)<=XL AND X2(SR,L)<=XL THEN 2130   'Linia jest widoczna
1080 IF X1(SR,L)>=XR AND X2(SR,L)>=XR THEN 2130   'Linia jest widoczna
1090  'W przeciwnym razie co najmniej jeden punkt linii jest we
1100  'wnętrzu prostokąta otoczenia tej ściany
1110  'Znajdź nachylenie i przesunięcie linii. Zapamiętaj w MT i BT
1120 IF X1(SR,L) = X2(SR,L) THEN 1170     'Linia pionowa
1130 IF Y1(SR,L) = Y2(SR,L) THEN 1190     'Linia pozioma
1140 MT = (Y1(SR,L) - Y2(SR,L)) / (X1(SR,L) - X2(SR,L))
1150 BT = Y1(SR,L) - MT * X1(SR,L)
1160 GOTO 1210
1170 MT = 9999    'Nachylenie linii pionowej nie jest zdefiniowane
1180 GOTO 1210
1190 MT = 0       'Nachylenie linii poziomej wynosi 0
1200 BT = Y1(SR,L)
```

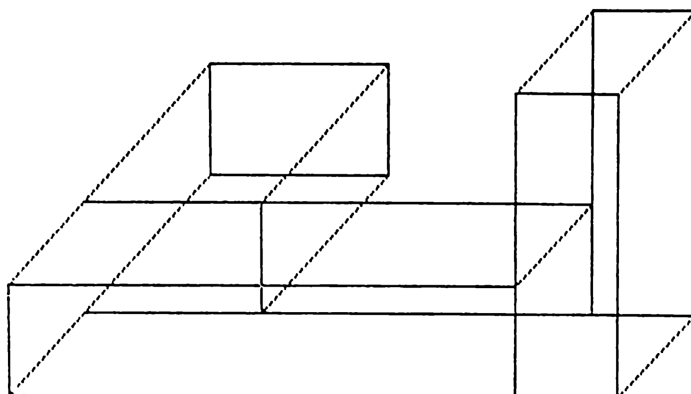
Program 9.4 (cd.)

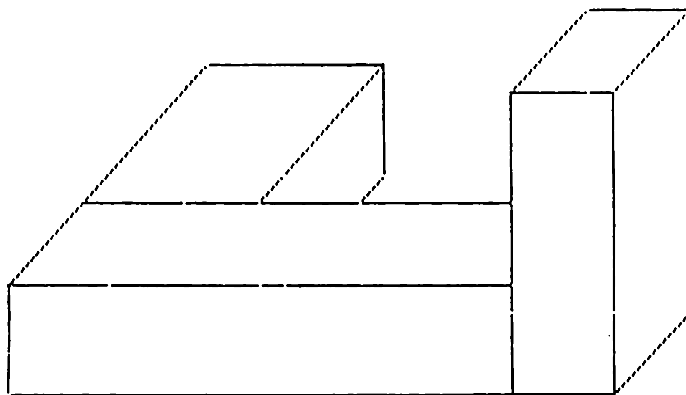
```
1210 C = 0          'C jest liczbą końców niewidocznego segmentu
1220    'Zbadaj widoczność w stosunku do wszystkich boków ściany
1230 FOR K = 1 TO NV(S)
1240   'Jeżeli ten bok jest wspólny, to przejdź do następnego
1250   IF X1(S,K)=X1(SR,L) AND Y1(S,K)=Y1(SR,L) AND X2(S,K)=X2(SR,L)
          AND Y2(S,K)=Y2(SR,L) THEN 2130
1260   IF X1(S,K)=X2(SR,L) AND Y1(S,K)=Y2(SR,L) AND X2(S,K)=X1(SR,L)
          AND Y2(S,K)=Y1(SR,L) THEN 2130
1270   IF M(K) = MT THEN 2090          'Linie są równoległe
1280   'Jeśli nie, znajdź punkt przecięcia (XP,YP) linii L badanej
1290   'ściany SR i linii K ściany S
1300   IF X1(S,K) <> X2(S,K) THEN 1340
1310   XP = X1(S,K)                    'Linia badana jest pionowa
1320   YP = MT * XP +BT
1330   GOTO 1490
1340   IF X1(SR,L) <> X2(SR,L) THEN 1380
1350   XP = X1(SR,L)                   'Linia badana jest pionowa
1360   YP = M(K) * XP +B(K)
1370   GOTO 1490
1380   IF Y1(S,K) <> Y2(S,K) THEN 1420
1390   YP = Y1S,K)                     'Linia badana jest pozioma
1400   XP = (YP - BT) / MT
1410   GOTO 1490
1420   IF Y1(SR,L) <> Y2(SR,L) THEN 1460
1430   YP = Y1(SR,L)                   'Linia badana jest pozioma
1440   XP = (YP - B(K)) / M(K)
1450   GOTO 1490
1460   XP = (BT - B(K)) / (M(K) -MT)
1470   YP = (MT * XP + BT)
1480   'Linie przecinają się w punkcie (XP,YP). Jeśli (XP,YP) nie leży
1490   'między końcami linii K ściany S, przejdź do następnej linii
1500   'L ściany SR
1510   IF XP < X1(S,K) AND XP < X2(S,K) THEN 2090
1520   IF XP > X1(S,K) AND XP > X2(S,K) THEN 2090
1530   IF YP < Y1(S,K) AND YP < Y2(S,K) THEN 2090
1540   IF YP > Y1(S,K) AND YP > Y2(S,K) THEN 2090
1550   '
1560   'W przeciwnym razie znajdź wartość wsp. Z dla punktu (XP,YP) na
1570   'linii badanej i na ścianie. Jeśli wartość Z dla (XP,YP) badanej
1580   'linii (ZL) jest mniejsza niż wartość Z dla (XP,YP) na linii  ściany
1590   '(ZS), to linia badana znajduje się z przodu ściany i jest widoczna
1600   IF X1(SR,L) = X2(SR,L) THEN 1630
1610   ZL=(XP-X1(SR,L))/(X2(SR,L)-X1(SR,L))*(Z2(SR,L)-Z1(SR,L))+Z1(SR,L)
1620   GOTO 1640
1630   ZL=(YP-Y1(SR,L))/(Y2(SR,L)-Y1(SR,L))*(Z2(SR,L)-Z1(SR,L))+Z1(SR,L)
1640   'Znajdź wartość współrzędnej Z punktu przecięcia
1650   IF X1(S,K) = X2(S,K) THEN 1680
1660   ZS = (XP-X1(S,K))/(X2(S,K)-X1(S,K))*(Z2(S,K)-Z1(S,K))+Z1(S,K)
1670   GOTO 1690
1680   ZS = (YP-Y1(S,K))/(Y2(S,K)-Y1(S,K))*(Z2(S,K)-Z1(S,K))+Z1(S,K)
1690   IF ZL < ZS THEN 2090
1700   '
1710   'Jeśli punkt (XP,YP) nie leży między końcami linii L ściany SR,
1720   'to punkt (XP,YP) musi być zamieniony na jeden z końców linii
1730   'L ściany SR
1740   IF XP < X1(SR,L) AND XP < X2(SR,L) THEN 1770
1750   IF XP > X1(SR,L) AND XP > X2(SR,L) THEN 1770
1760   GOTO 1840
1770   IF ABS(XP - X1(SR,L)) < ABS(XP - X2S(R,L)) THEN 1810
1780   XP = X2(SR,L)
```

Program 9.4 (cd.)

```
1790   YP = Y2(SR,L)
1800   GOTO 1970
1810   XP = X1(SR,L)
1820   YP = Y1(SR,L)
1830   GOTO 1970
1840   IF YP < Y1(SR,L) AND YP < Y2(SR,L) THEN 1870
1850   IF YP > Y1(SR,L) AND YP > Y2(SR,L) THEN 1870
1860   GOTO 1970
1870   IF ABS(YP - Y1(SR,L)) < ABS(YP - Y2S(R,L)) THEN 1910
1880   YP = Y2(SR,L)
1890   XP = X2(SR,L)
1900   GOTO 1970
1910   YP = Y1(SR,L)
1920   XP = X1(SR,L)
1930   '
1940   'Czyżbyśmy przypadkowo wygenerowali linię brzegową? Jeśli linia
1950   'badana ma takie samo nachylenie i punkt wspólny z jakąś linią
1960   'ściany, to nie wymazuj jej.
1970   FOR J = 1 TO NV(S)
1980     IF MT = M(J) AND YP = M(J) *XP + B(J) THEN 2130
1990   NEXT J
2000   IF C <> 0 THEN 2050
2010   XA = XP                    'To jest pierwszy punkt ukrytego segmentu
2020   YA = YP                    'zapamiętany w XA, YA
2030   C = C + 1
2040   GOTO 2090
2050   IF XP = XA AND YP = YA THEN 2090   'Ten sam punkt (wierzchołek)
2060   XD = XP                             'Zapamiętaj drugi punkt w XD, YD
2070   YD = YP
2080   C = C + 1
2090 NEXT K
2100   '
2110 IF C <> 2 THEN 2130
2120 DRAWLINE XA,YA TO XD,YD              'Usuń niewidoczny fragment linii
2130 RETURN
2140 '**********************************************************
2150 DATA 14
2160 DATA 4,30,130,67,30,170,71,230,170.21.230,130.17
2170 DATA 4,230,60,10,230.170,21.270,170,11,270,60,0
2180 DATA 4,270,60,0,270,170,11,300,140,61,300,30,50
2190 DATA 4,260,30,60,260,140,71,300,140,61.300,30,50
2200 DATA 4,130,100,99.5,130,140,103.5,260,140,71,260,100,67
2210 DATA 4,130,100,99.5.130.140,103.5,180,90,186.8,180.50.182.8
2220 DATA 4,110,50,200.3,110,90,204.3,180,90.186.8.180,50.182.8
2230 DATA 4.30,130,67.30,170.71,110.90,204.3,110.50,200.3
2240 DATA 4,230,60,10.230,130.17.260,100,67.260,30,60
2250 DATA 4,260,30.60,230,60,10,270,60.0,300,30.50
2260 DATA 4,110,50,200.3,60,100,117.130,100.99.5,180,50,182.8
2270 DATA 4,60,100.117.30,130.67,230,1310,7,260,100,67
2280 DATA 4,110,90,204.3,60,140,121.130,140,103.5,180,90,186.8
2290 DATA 4,60,140,121,30,170,71.270.170,11,300.140.61
2300 END
```

14 powierzchni o 4 wierzchołkach każda. Powierzchnie te mogą stanowić część obiektu (tak jak w przykładzie) lub mogą reprezentować różne obiekty. Izolowane linie proste wprowadza się jako powierzchnie o 2 wierzchołkach. Po wprowadzeniu wszystkich powierzchni program narysuje wszystkie linie (rys. 9.11a), a następnie znajdzie i usunie każdy fragment linii ukryty za powierzchnią (rys. 9.11b). W celu


(a)


(b)

Rys. 9.11. Obiekt trójwymiarowy przed (a) i po usunięciu (b) linii niewidocznych narysowany przez program 9.4

uniknięcia wielokątów wklęsłych, niektóre powierzchnie obiektów zostały przekształcone tak, aby utworzyć dwa wielokąty wypukłe.

Każdą wprowadzoną do programu 9.4 powierzchnię wybiera się po kolei, a następnie bada widoczność wszystkich pozostałych linii w stosunku do tej powierzchni. Jeśli linia leży poza prostokątnym otoczeniem powierzchni, to jest widoczna. W przeciwnym razie trzeba sprawdzić jej widoczność w stosunku do rzeczywistych granic wielokąta. W tym celu oblicza się punkt jej przecięcia z każdym bokiem powierzchni. Linia badana i bok przecinają się, jeśli punkt przecięcia leży między koń-

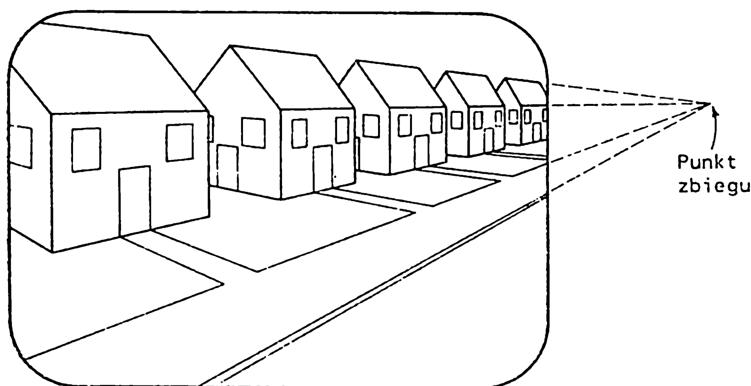
cami każdej z nich. W tym przypadku sprawdzamy, czy wartość Z linii badanej jest większa niż wartość Z boku w punkcie przecięcia. Jeśli tak jest, to zapamiętuje się ten punkt jako jeden koniec niewidocznego fragmentu linii. Jeśli punkt przecięcia leży między końcami boku powierzchni, ale nie między końcami linii badanej, to zapamiętujemy bliższy koniec linii badanej jako jeden z końców niewidocznego fragmentu linii. W pozostałych przypadkach nic się nie zapamiętuje. Gdy tą metodą znajdzie się dwa punkty na linii badanej, wówczas usuwa się odcinek między nimi. Jeśli nie można znaleźć dwóch punktów, to linia nie jest zakryta przez powierzchnię.

Program można rozszerzyć o możliwość rozważenia powierzchni wklęsłych i bardziej złożonych możliwości przecięć — linia może przecinać powierzchnię wklęsłą w więcej niż dwóch punktach. W tej sytuacji należy określić, czy końce linii leżą wewnątrz czy poza wielokątem i zaczynając z jednego końca określać niewidoczne fragmenty. Jeśli wybrany koniec linii leży wewnątrz wielokąta, to wymazujemy odcinek linii od tego końca do pierwszego przecięcia z bokiem. Usuwamy także odcinek od drugiego przecięcia do trzeciego itd. Jeśli wybrany koniec linii leży na zewnątrz wielokąta, to wymazywanie dotyczy odcinka między pierwszym a drugim przecięciem, między trzecim a czwartym itd. Linia może być także widoczna częściowo, jeśli przecina powierzchnię w punkcie wewnętrznym tej powierzchni. Może tak się zdarzyć, jeśli linia lub powierzchnia mają zmienne wartości Z i jeden koniec linii znajduje się przed powierzchnią, a drugi za nią. W takim przypadku można określić punkt przecięcia posługując się równaniami linii i powierzchni w trzech wymiarach, ale program jest bardziej złożony. Przedyskutowano zaledwie kilka z wielu możliwych metod usuwania niewidocznych linii i powierzchni. Niektóre z nich nadają się tylko do pewnych kształtów powierzchni. Niektóre wymagają większej pojemności pamięci lub dłuższego czasu obliczeń. W miarę wzrostu złożoności kształtów obiektów w obrazie zwiększa się złożoność metod wymazywania. Zwykle dążymy do znalezienia metody, która usuwa niewidoczne linie w najkrótszym czasie. Jest to szczególnie ważne w zastosowaniach z animacją obrazów.

9.4. Rzuty perspektywiczne

Usunięcie niewidocznych linii w scenach trójwymiarowych dostarcza wizualnej informacji o głębi sceny i zwiększa realizm jej płaskiej reprezentacji. Dodatkową metodą osiągnięcia realizmu i głębi sceny jest projekcja obiektów na ekranie w *perspektywie*. Oglądając obiekty naturalne stwierdzamy, że wydają się one mniejsze, gdy znajdują się w większej odległości od nas. W rzędzie budynków przedstawionym na rys. 9.12 wydaje się, że budynki położone bliżej są większe niż położone dalej. W rzucie perspektywicznym linie równoległe zbliżają się do siebie tym bardziej, im są bardziej oddalone od obserwatora.

W celu otrzymania rzutu perspektywicznego można scenę z rys. 9.12 rozplanować na papierze milimetrowym tak, aby wszystkie linie równoległe przecinały się w pewnym punkcie zwanym *punktem zbiegu*. Umożliwia to pomiar współrzędnych X i Y obiektów oraz ich wykreślenie. W ten sposób można wyświetlić w rzucie perspektywicznym jakąś scenę narysowaną na papierze milimetrowym. Jednakże każda zmiana w scenie oraz każda nowa scena wymagają powtórzenia całego procesu planowania na papierze milimetrowym.



Rys. 9.12. W rzucie perspektywicznym linie równoległe schodzą się w punkcie zbiegu, wskutek czego obiekty odległe wydają się mniejsze niż obiekty położone blisko

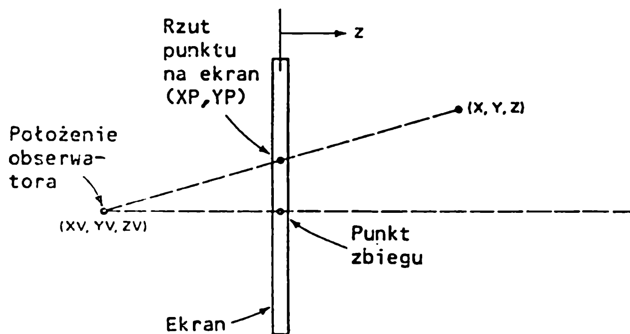
Inna metoda otrzymania rzutu perspektywicznego polega na zastosowaniu w programie wyświetlania równań perspektywy, co zapewnia większą elastyczność. W perspektywie można wtedy przedstawiać sceny animowane oraz umieszczać powtarzające się wzory bez potrzeby planowania sceny dla każdego pojawienia się wzoru. Jednokrotne określenie współrzędnych jednego budynku z rys. 9.12 umożliwia wielokrotne wykreślenie go w wymiarach odpowiadających jego położeniu.

Dla każdego punktu (X, Y, Z) w scenie trójwymiarowej współrzędne punktu po transformacji (XP, YP) na ekranie określa się jako

$$\begin{aligned} XP &= XV + (XV - X) * ZV/(Z - ZV) \\ YP &= YV + (YV - Y) * ZV/(Z - ZV) \end{aligned} \qquad (9.1)$$

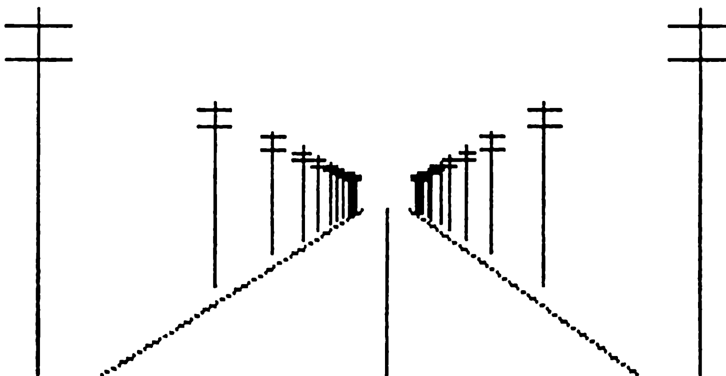
przy czym punkt (XV, YV, ZV) jest *punktem patrzenia* (położeniem obserwatora) umieszczonym przed ekranem. Punkt (XV, YV) jest punktem zbiegu na ekranie. Na rysunku 9.13 zilustrowano zależności między różnymi wartościami współrzędnych punktów przy bocznym widoku ekranu. Dla punktów o wartości Z = 0 nie wystąpi zmiana współrzędnych: (XP, YP) = (X, Y). Dla punktów z tyłu ekranu (Z > 0) projekcja na ekranie jest tym bliżej punktu zbiegu, im większe jest Z. Współrzędna ZV musi być zawsze ujemna, gdyż scenę oglądamy z przodu ekranu. Większe

wartości ZV dają gorszą perspektywę (mniejszą zbieżność linii równoległych). Zbliżając punkt patrzenia do ekranu zwiększamy zniekształcenia perspektywiczne — zwiększa się zbieżność linii równoległych.



Rys. 9.13. Punkt o współrzędnych (X, Y, Z) rzutowany na ekran ma współrzędne (XP, YP), gdy obserwator znajduje się w punkcie o współrzędnych (XV, YV, ZV)

W programie 9.5 obliczamy i wyświetlamy rzut perspektywiczny drogi z linią telefoniczną napowietrzną – wynik przedstawiono na rys. 9.14. Program demonstruje technikę wielokrotnego kreślenia zadanego obiektu (w tym przypadku słupa telefonicznego) dla różnych jego położeń i obliczania przesuniętych pozycji obiektu na podstawie równań perspektywy (9.1). Położenie punktu zbiegu i wartość ZV możemy zmieniać dopóty, dopóki obiekty nie są rzutowane poza ekran.



Rys. 9.14. Widok perspektywiczny drogi ze słupami telefonicznymi – wynik programu 9.5

Program 9.5. Rysowanie trójwymiarowej sceny z powtarzającym się rysunkiem obiektu w perspektywie (słup telefoniczny)

```
10  'PROGRAM 9.5. Słupy telefoniczne w perspektywie
20      'Słupy telefoniczne sa zdefiniowane w instrukcjach DATA.
30      'Z klawiatury wprowadza sie położenie punktu patrzenia.
40      'Z równan perspektywy oblicza się nowe wartości współrzed-
50      'nych X, Y dla różnych wartości Z.
60  CLEARSCREEN
70  DIM X1(8), Y1(8), X2(8), Y2(8)
80  XM = 279
90  YM = 169            'XM i YM są maksymalnymi wartościami dla systemu
100     'Ustal współrzędne punktu patrzenia
110 PRINT "Wprowadź współrzędne X, Y, Z punktu patrzenia"
120 PRINT "X i Y muszą być na ekranie. Z musi być ujemne"
130 INPUT XV, YV, ZV
140 IF XV < 0 OR XV > XM OR YV < 0 OR YV > YM THEN 110
150 IF ZV => 0 THEN 110
160 CLEARSCREEN
170 GRAPHICS
180 FOR K = 1 TO 6
190   READ X1, Y1, X2, Y2
200   FOR Z = 0 TO 5000 STEP 500
210     'Oblicz stałą część równania
220     P = -ZV / (Z - ZV)
230     'Oblicz punkty linii dla tej wartości Z
240     XA = XV + (X1 - XV) * P
250     YA = YV + (Y1 - YV) * P
260     XB = XV + (X2 - XV) * P
270     YB = YV + (Y2 - YV) * P
280     DRAWLINE XA,YA TO XB,YB
290   NEXT Z
300 NEXT K
310     'Narysuj drogę
320 FOR K = 1 TO 3
330   READ XA, YA
340   XB = XV + (XA - XV) * P
350   YB = YV + (YA - YV) * P
360   DRAWLINE XA,YA TO XB,YB
370 NEXT K
380     '****************************************
390 DATA 50,45,50,155
400 DATA 40,60,60,60
410 DATA 40,50,60,50
420 DATA 260,45,260,155
430 DATA 250,60,270,60
440 DATA 250,50,270,50
450 DATA 70,155,160,155,240,155
460 END
```

Rzut perspektywiczny jednego obiektu można utworzyć za pomocą programu 9.6. Na środku ekranu jest wyświetlany rysunek będący rzutem prostokątnym figury. Wybierając różne położenia obserwatora, możemy wyświetlić różne rzuty perspektywiczne obiektu (rys. 9.15). Program ten pokazuje ogólną metodę definiowania obiektu trójwymiarowego we współrzędnych ekranu i obliczania rzutu obiektu przy korzystaniu z równań perspektywy.

Program 9.6 identyfikuje powierzchnie ukryte przez obliczanie odległości od punktu patrzenia (położenia obserwatora). Ponieważ rozważane w przykładzie

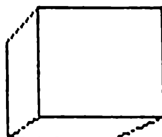
Program 9.6. Rzut perspektywiczny obiektu trójwymiarowego (pudełka)

```
10  'PROGRAM 9.6. Obiekt trójwymiarowy w perspektywie
20      'Program definiuje trówymiarowy obiekt, ustala punkt, z którego
30      'obiekt będzie oglądany, oblicza współrzędne punktów obiektu
40      'z pomocą równań perspektywy i rysuje widoczną część obiektu.
50      'Widoczność ścian obiektu określa się korzystając z zasad
55      'symetrii.
60      '************************************************************
70  CLEARSCREEN
80  DIM X(8), Y(8), Z(8), XP(8), YP(8), D(8)
90  XM = 255
100 YM = 191
110     '************* Odczytaj dane do rysunku ***************
120 READ N
130 FOR K = 1 TO N
140   READ X(K), Y(K), Z(K)
150   IF X(K)<0 OR X(K)>XM OR Y(K)<0 OR Y(K)>YM THEN 880
160 NEXT K
170 GRAPHICS
180 FOR K = 1 TO N/2 - 1       'Narysuj ścianę przednią
190   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
200 NEXT K
210 DRAWLINE X(N/2),Y(N/2) TO X(1),Y(1)
220     '********** Ustal punkt patrzenia i rysuj **************
230 POSITION 1,1
240 PRINT "Wprowadź współrzędne punktu patrzenia"
250 PRINT "Współrzędna Z musi być ujemna"
260 PRINT "Napisz 0,0,0 aby zakończyć program"
270 INPUT XV, YV, ZV
280 IF XV = 0 AND YV = 0 AND ZV = 0 THEN 880
290 IF ZV >= 0 THEN 240
310     'Oblicz perspektywę punktów
320 FOR K = 1 TO N
330   XP(K) = XV + (X(K) - XV) * (-ZV) / (Z(K) - ZV)
340   YP(K) = YV + (Y(K) - YV) * (-ZV) / (Z(K) - ZV)
350 NEXT K
360     'Znajdź odległości punktów od punktu patrzenia
370 FOR K = 1 TO N
380   D(K) = SQR ((X(K)-XV)^2 + (X(K)-YV)^2 + (Z(K)-ZV)^2)
390 NEXT K
400 GOSUB 420                  'Rysuj
410 GOTO 220
420     '**************** Program rysowania ********************
430 CLEARSCREEN
440 IF D(1) = D(5) THEN 570
450 IF D(1) > D(5) THEN 520
460 IF Z(1) <= Z(V) THEN 570   'Czy ściana jest widoczna?
470 FOR K = 1 TO 3             'Narysuj ścianę zawierającą punkt 1
480   DRAWLINE XP(K),YP(K) TO XP(K+1),YP(K+1)
490 NEXT K
500 DRAWLINE XP(4),YP(4) TO XP(1),YP(1)
510 GOTO 570
520 IF Z(5) >= ZV THEN 570     'Czy ściana jest widoczna?
530 FOR K = 5 TO 7             'Narysuj ścianę zawierającą punkt 5
540   DRAWLINE XP(K),YP(K) TO XP(K+1),YP(K+1)
550 NEXT K
560 DRAWLINE XP(8),YP(8) TO XP(5),YP(5)
570 IF D(1) = D(4) THEN 700
580 IF D(1) > D(4) THEN 650
590 IF X(1) >= XV  THEN 700             'Czy ściana jest widoczna?
600 DRAWLINE XP(1),YP(1) TO XP(2),YP(2)  'Narysuj ścianę zaw. punkt 1
610 DRAWLINE XP(2),YP(2) TO XP(6),YP(6)
620 DRAWLINE XP(6),YP(6) TO XP(5),YP(5)
630 DRAWLINE XP(5),YP(5) TO XP(1),YP(1)
```

Program 9.6 (cd.)

```
640 GOTO 700
650 IF X(4) <= XV THEN 700                    'Czy ściana jest widoczna?
660 DRAWLINE XP(4),YP(4) TO XP(3),YP(3)       'Narysuj ścianę zaw. punkt 4
670 DRAWLINE XP(3),YP(3) TO XP(7),YP(7)
680 DRAWLINE XP(7),YP(7) TO XP(8),YP(8)
690 DRAWLINE XP(8),YP(8) TO XP(4),YP(4)
700 IF D(1) = D(2) THEN 830
710 IF D(1) > D(2) THEN 780
720 IF Y(1) <= YV  THEN 830                   'Czy ściana jest widoczna?
730 DRAWLINE XP(1),YP(1) TO XP(4),YP(4)       'Narysuj ścianę zaw. punkt 1
740 DRAWLINE XP(4),YP(4) TO XP(8),YP(8)
750 DRAWLINE XP(8),YP(8) TO XP(5),YP(5)
760 DRAWLINE XP(5),YP(5) TO XP(1),YP(1)
770 GOTO 830
780 IF Y(2) >= YV  THEN 830                   'Czy ściana jest widoczna?
790 DRAWLINE XP(2),YP(2) TO XP(3),YP(3)       'Narysuj ścianę zaw. punkt 2
800 DRAWLINE XP(3),YP(3) TO XP(7),YP(7)
810 DRAWLINE XP(7),YP(7) TO XP(6),YP(6)
820 DRAWLINE XP(6),YP(6) TO XP(2),YP(2)
830 RETURN
840    '************************************************************
850 DATA 8
860 DATA 150,40,0,150,80,0,100,80,0,100,40,0
870 DATA 150,40,30,150,80,30,100,80,30,100,40,30
880 END
```

```
Wprowadź współrzędne punktu patrzenia
Współrzędna Z musi być ujemna
Napisz 0,0,0 aby zakończyć program
```



Rys. 9.15. Rzut perspektywiczny trójwymiarowego obiektu, widzianego z dołu i z lewej strony wyświetlony za pomocą programu 9.6

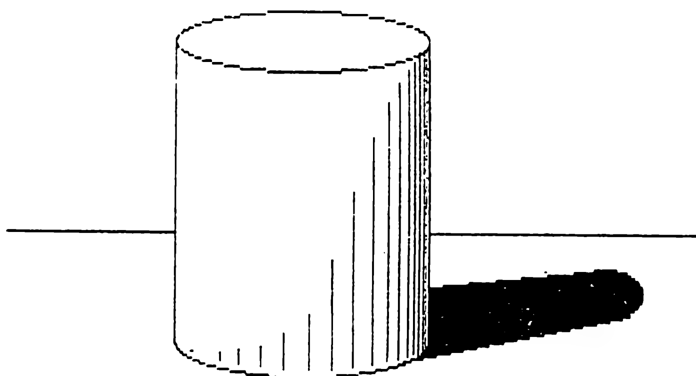
pudełko jest symetryczne, sprawdza się odległości od przeciwległych ścian. Ta ściana z pary, która jest położona dalej od obserwatora, nie jest widoczna. Odległość punktu (X, Y, Z) od punktu (XV, YV, ZV) oblicza się jako

$$D = SQR((X - XV)^2 + (Y - YV)^2 + (Z - ZV)^2) \qquad (9.2)$$

Następnie sprawdza się widoczność wybranej ściany, gdyż może wystąpić taka sytuacja, że żadna ściana (z pary) nie jest widoczna. Na przykład, przy patrzeniu na sześcian bezpośrednio z przodu widać tylko jedną ścianę. Ściany boczne, góra lub dół będą widoczne tylko wówczas, gdy punkt patrzenia zostanie przesunięty. W przypadku obiektów niesymetrycznych możemy korzystać z metody stosowanej w programie 9.4 i zależności (9.2) do obliczania odległości.

9.5. Cieniowanie i rozjaśnianie

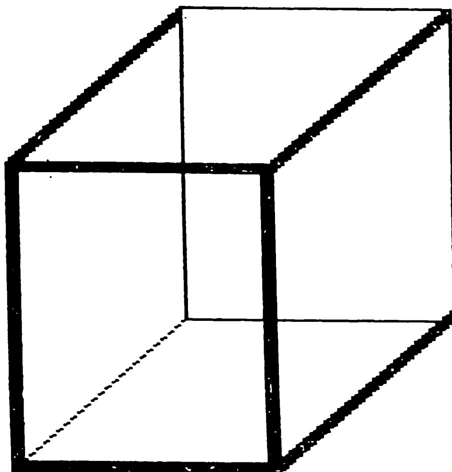
Informację o głębi sceny trójwymiarowej i jej realizm można poprawić stosując cieniowanie. Na rysunku 9.16 pokazano przykład wzoru cieniowania dla powierzchni krzywoliniowych. Wzory cieniowania ułatwiają percepcję konturów i krzywizny powierzchni. Obszary ciemniejsze i cienie pomagają ponadto zidentyfikować tylne części obiektów. W celu dodania cieni wybieramy położenie źródła światła, na przykład z lewej strony i z przodu ekranu, a następnie cieniujemy strony naprzeciw źródła światła. Na ogół delikatne wzory cieniowania i cienie nie są skuteczne, jeśli rozdzielczość systemu graficznego nie jest dostateczna. Natomiast proste cieniowanie ścian naprzeciw źródła światła może być skuteczne nawet przy małej rozdzielczości.



Rys. 9.16. Wzór cieniujący i cień pogłębiają realizm trójwymiarowej sceny

Niektóre ze wzorów cieniowania omawianych w rozdz. 3 definiuje się w postaci podprogramów i używa w programie wyświetlania w taki sam sposób, jak w programie 9.1 definiowano wnętrze prostokąta. Najpierw określa się granice obszaru cieniowanego, a następnie nakłada się na ten obszar wzór cieniujący (zamiast wymazywania jak w programie 9.1). Takie podprogramy noszą nazwę *maski cieniującej* i mogą służyć do cieniowania różnych wielokątów, obszarów kolistych lub eliptycznych.

Rozjaśnienie może być stosowane jako alternatywa usuwania niewidocznych linii w scenach trójwymiarowych. Przy tej metodzie dodajemy informację o głębi sceny przez identyfikację przednich linii obiektu. Realizuje się to, sortując linie ze względu na wartość współrzędnej Z i wyróżniając linie położone bliżej jaśniejszym kolorem lub podwójną grubością. Na rysunku 9.17 pokazano przykład wyróżnienia linii przednich przez wykreślenie ich jako grubsze niż linie tylne, niewidoczne. Dla obiektów wyświetlanych w postaci konturu rozjaśnianie może być szybką metodą odróżnienia przedniej strony obiektów od tylnej.



Rys. 9.17. Rozjaśnione linie obiektu trójwymiarowego służą do identyfikacji ścian położonych bliżej

9.6. Wykresy

Skuteczną metodą prezentacji różnorodnych zależności mogą być wykresy trójwymiarowe. Trójwymiarowy wykres słupkowy utworzony przez program 9.7 pokazuje relacje między trzema zmiennymi: zmianami liczby ludności dwóch miast w ciągu trzech dekad. W programie korzystano z metod opisanych przy prezentacji pro-

Program 9.7. Trójwymiarowy wykres słupkowy

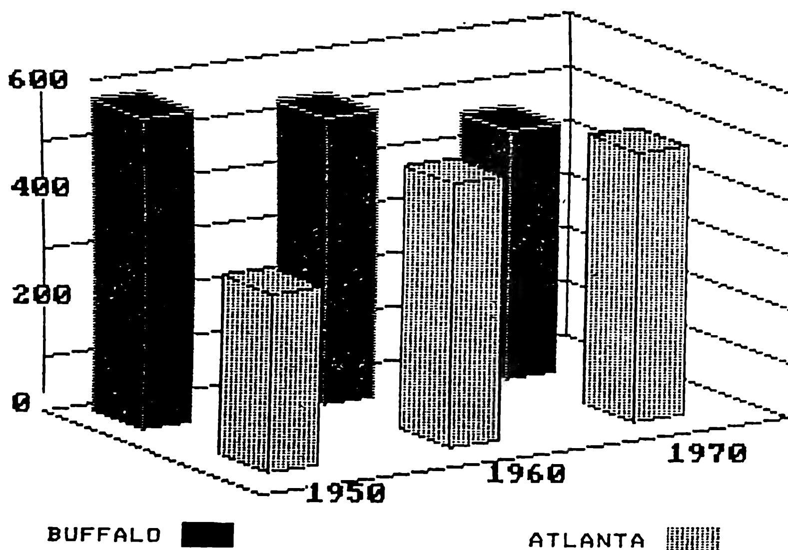
```
10  'PROGRAM 9.7. Trójwymiarowy wykres słupkowy
20  CLEARSCREEN
30  GRAPHICS
40      '********************* Narysuj tlo ***********************
50  X = 230
60  XL = 20
70  XR = 319
80  DRAWLINE X,0 TO X,120
90  DRAWLINE XL,30 TO XL,150
100 DRAWLINE XR,30 TO XR,150
110 A = 600                          'A jest liczbą do opisu osi
120 RO = 4
130 C$ = "E"         'C$ jest wskaźnikiem parzystości rzędu (E -parzysty,
140 FOR Y = 0 TO 120 STEP 120/6                  'O - nieparzysty)
150   DRAWLINE X,Y TO XL,Y+30
160   'Opisz oś tylko w parzystych rzędach
170   IF C$ = "O" THEN 240
180   POSITION RO,1
190   PRINT A
200   C$ = "O"
210   A = A - 200
220   RO = RO + 5
230   GOTO 250
240   C$ = "E"
250   DRAWLINE X,Y TO XR,Y+30
```

Program 9.7 (cd.)

```
260 NEXT Y
270   'Narysuj bazę
280 DRAWLINE 20,150 TO 109,180
290 DRAWLINE 109,180 TO 319,150
300 POSITION 23,17:PRINT "1950"
310 POSITION 22,26:PRINT "1960"
320 POSITION 21,35:PRINT "1970"
330 POSITION 25,1
340 PRINT "  BUFFALO                  ATLANTA";
350 ML = -30/210             'ML jest nachyleniem kratki z lewej strony
360 MR = 30/89               'MR jest nachyleniem kratki z prawej strony
370 H = 40                 'H jest poziomym wymiarem każdego słupka
380 C = 0                  'C jest liczbą miast na wykresie
390 XL = 40                'Dolny, lewy róg słupka w punkcie XL,YL
400 YL = 150
410 F = 3                  'F jest kolorem wypełnienia słupka
420 O = 2                  'O jest kolorem obrysu słupka
430 XS = 78
440     '
450     'Niech kolor kodu miasta będzie czarny
460 COLOR F,0              'Użyj koloru wypełniania na czarnym tle
470 FOR S = XS TO XS+20
480   DRAWLINE X,191 TO X,199
490 NEXT K
500     '************** Narysuj słupki dla miast ****************
510 FOR K = 1 TO 3
520   READ V
530   'Przeskaluj liczbę ludności do wymiarów ekranu
540   HT = (120 * V) / 600
550   GOSUB 700             'Zapełnij kolorem obszar słupka
560   GOSUB 970             'Narysuj obrys słupka
570   XL = XL + 73          'Przejdź do następnego słupka
580   YL = YL - 10
590 NEXT K
600 C = C +1
610 IF C = 2 THEN 1080      'Czy już koniec?
620   'Podstaw początkowe wartości zmiennych dla następnego miasta
630 XL = 91
640 YL = 165
650 F = 5                   'Kolor słupka dla następnego miasta
660 O = 4                   'Kolor obrysu dla następnego miasta
670 XS = 272
680 GOTO 450
690 '
700 '***************** Zapełnij obszar slupka ********************
710 BL = YL - HT -ML * XL     'BL jest przecięciem lewgo brzegu słupka
720 BR = YL - ML * XL         'BR jest przecięciem prawego brzegu słupka
730 FOR X = XL TO XL+H/2
740   Y1 = ML * X + BL
750   Y2 = MR * X + BR
760   COLOR F,0
770   DRAWLINE X,Y1 TO X,Y2
780   COLOR O,0
790   POINTPLOT X,Y1
800   POINTPLOT X,Y2
810 NEXT K
820 YT = Y1                   'YT i YB oznaczają górę i dół słupka
830 YB = Y2
840 BL=YB-ML*(XL+H/2)  'BL - przesunięcie prawego, dolnego brzegu słupka
850 BR=YT-MR*(XL+H/2)  'BR - przesunięcie prawego, górnego brzegu słupka
860 FOR X = XL+H/2 TO XL+H
870   Y1 = MR * X + BR
880   Y2 = ML * X + BL
890   COLOR F,0
```

Program 9.7 (cd.)

```
900   DRAWLINE X,Y1 TO X,Y2
910   COLOR 0,0
920   POINTPLOT X,Y1
930   POINTPLOT X,Y2
940 NEXT X
950 YR = Y2
960 RETURN
970 '*************** Narysuj obrys słupka ********************
980  COLOR 0,0         'Podaj kolor obrysu
990  DRAWLINE XL,YL TO XL,YL-HT
1000 DRAWLINE XL+H/2,YB TO XL+H/2,YB-HT
1010 DRAWLINE XL+H,YR TO XL+H,YR-HT
1020 DRAWLINE XL,YL-HT TO XL+H/2,YB-HT
1030 DRAWLINE XL+H/2,YB-HT TO XL+H,YR-HT
1040 RETURN
1050 '***********************************************************
1060 DATA 580,532,462
1070 DATA 331,487,498
1080 END
```



Rys. 9.18. Trójwymiarowy wykres słupkowy pokazujący liczbę ludności (w tysiącach) dla dwóch miast w latach 1950, 1960, 1970 — narysowany przez program 9.7

gramu 9.1. Najpierw wykreśla się powierzchnie dalsze tak, że gdy kreśli się powierzchnie bliższe, zakrywają one te dalsze i wymazują je. Wysokość słupków jest obliczana w odpowiedniej skali według metod przedstawionych w rozdz. 4. Aby ułatwić spostrzeganie wysokości i położenia słupków, powiększono odległości między nimi. Program 9.7 można rozszerzyć włączając dodatkowe zmienne. Na przykład możemy wykreślić każdy słupek jako całkowitą liczbę ludności, a następnie podzielić go na

części odpowiadające różnym grupom wiekowym. Te części można następnie pokolorować wg ustalonego kodu kolorów. Istnieje także wiele sposobów opisywania. Można wymyślić takie napisy, które będą nachylone zgodnie z kierunkiem słupków albo można je pokolorować lub pocieniować i nanieść na obraz, tak jak to pokazano na rys. 9.18.

W celu zilustrowania zależności między wieloma zmiennymi w trzech wymiarach używa się linii krzywych. Na przykład program 9.8 rysuje wykres funkcji

$$Y = YC + H * SIN(W * SQR(X * X + Z * Z))$$

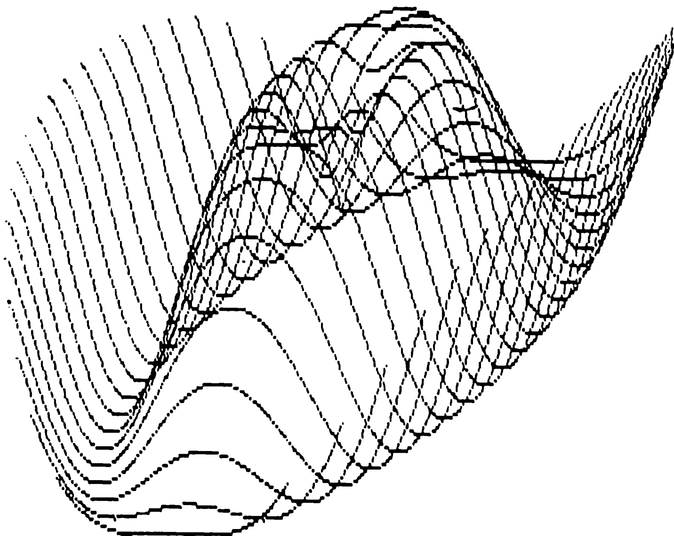
dla wybranego zakresu zmiennych X i Z i wybranych wartości stałych YC, H i W. Zakres wartości X obliczamy z zależności

$$XR = SQR(1 - (Z * Z)/(ZR * ZR))$$

Ta funkcja powoduje, że zakres zmian X zależy od każdorazowej wartości Z. Zaczynamy więc od kreślenia krótkich krzywych (Z ma wartość bliską minimalnej i równą −ZR), zwiększamy ich długość do wartości maksymalnej (Z = 0), a następnie zmniejszamy do wartości początkowej (Z jest bliskie ZR). Kształt tej powierzchni pokazano na rys. 9.19. W celu określenia dopuszczalnego zakresu zmian X i Z korzysta się z innych relacji (takich, jak funkcja liniowa lub stała). Zakres Z i stałe XC, YC, H, W i XR zostały tak wybrane, aby zapełnić rysunkiem ekran. W pro-

Program 9.8. Kreślenie linii krzywych w trzech wymiarach

```
10  'PROGRAM 9.8. Rysunek trójwymiarowy
20      'Program rysuje funkcję SIN w trzech wymiarach. Współrzędna
30      'Z zmienia się od -ZR do ZR. Dla każdej wartości Z znajduje
40      'się zakres wartości X. Zmieniając X w tym zakresie oblicza
50      'się wartości Y i rysuje punkt o współrzędnych X,Y. Wartości
60      'ZR i XR są tak wybrane, aby wykres mieścił się na ekranie
70      'o wymiarach 640 x 200 pikseli.
80      '**********************************************************
90  CLEARSCREEN
100 GRAPHICS
110 XC = 320
120 YC = 115
130 XR = 175
140 ZR = 120                    'Z zmienia się w przedziale od -ZR do +ZR
150 H =40
160 W = 0.043
170 FOR Z = -ZR+1 TO ZR-1 STEP 10
180   XL = INT(XR * SQR(1 -(Z *Z) / (ZR * ZR)) +0.5)
190   X = -XL
200   Y = H * SIN(W *SQR(X * X + Z * Z))
210   X1 = XC + X + Z
220   Y1 = 199 - (YC + Y + Z/2)
230   FOR X = -XL+1 TO XL-1 STEP 5
240     Y = H * SIN(W * SQR(X * X + Z * Z))
250     X2 = XC + X + Z
260     Y2 = 199 - (YC + Y + Z/2)
270     DRAWLINE X1,Y1 TO X2,Y2
280     X1 = X2
290     Y1 = Y2
300   NEXT X
310 NEXT Z
320 END
```



Rys. 9.19. Linie krzywe w trzech wymiarach — narysowane przez program 9.8

gramie 9.8 przyjęto wymiary ekranu 640×200 pikseli. Przez wykreślenie X + Z i Y + Z/2, zamiast odpowiednio X i Y, przesuwamy każdą krzywą nieco w prawo i w górę ekranu, co daje efekt trójwymiarowości.

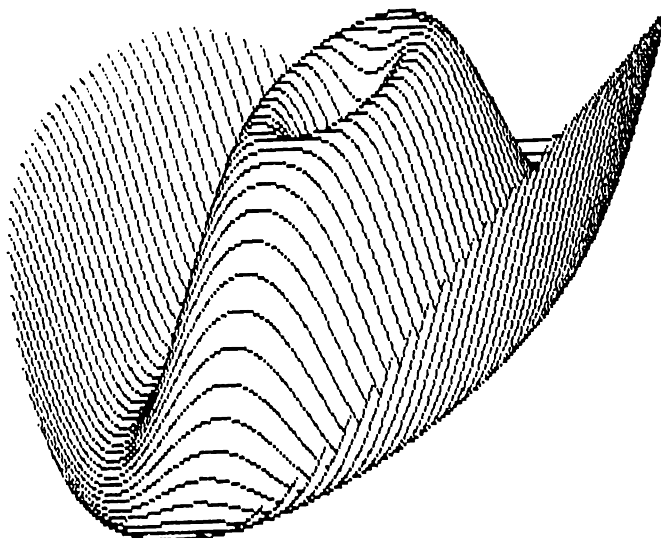
Program 9.9 jest modyfikacją programu 9.8; wyeliminowano w nim linie pokrywające się, a efekt przedstawiono na rys. 9.20. Wymazywanie niewidocznych

Program 9.9. Kreślenie linii krzywych w trzech wymiarach i wyświetlanie tylko widocznych fragmentów w celu nadania rysunkowi wyglądu powierzchni

```
10  'PROGRAM 9.9. Rysunek widocznej części wykresu trójwymiarowego
20      'Program rysuje funkcję SIN w trzech wymiarach. Współrzędna
30      'Z zmienia się od -ZR do ZR. Dla każdej wartości Z znajduje
40      'się zakres wartości X. Zmieniając X w tym zakresie oblicza
50      'się wartości Y przesuniętego punktu (X,Y). Dla każdej
60      'wartości X pamięta się górną i dolną granicę Y, określające
70      'zakres użytych już wartości Y. Każdą nowoobliczoną wartość Y
80      'porównuje się z granicami górną i dolną - jeśli mieści się
90      'między nimi, to jest pomijana. Jeśli jest poza nimi, to
100     'rysujemy punkt i zmieniamy wartości granic.
110     'Wartości XR i ZR są tak wybrane, aby wykres mieścił się
120     'na ekranie o wymiarach 640 x 200 pikseli.
130     '
140     '**********************************************************
150 CLEARSCREEN
160 DIM UB(424), LB(424)            'Wykres dla 424 wartości X
170 XC = 320
180 YC = 115
190 XR = 175
200 ZR = 120
210 H = 40
220 W = 0.043
```

Program 9.9 (cd.)

```
230 XA = 107
240     'Zadaj początkowe wartości w tablicach
250 FOR S = 1 TO 424
260   UB(S) = 0
270   LB(S) = 1000
280 NEXT S
290 GRAPHICS
300 FOR Z = -ZR+1 TO ZR-1 STEP 5
310   XL = INT(XR * SQR(1 -(Z * Z) / (ZR * ZR) + 0.5)
320   X = - XL
330   Y = H * SIN(W * SQR(X * X + Z * Z))
340   X1 = X + XC + Z
350   Y1 = INT(199 - (YC + Y + Z/2) + 0.5)
360   FOR X = -XL+1 TO XL-1
370     Y = H * SIN(W * SQR(X * X + Z * Z))
380     X2 = XC + X + Z
390     Y2 = INT(199 - (YC + Y + Z/2) + 0.5)
400     'Czy punkt jest wewnątrz granic tego, co już jest narysowane?
410     IF Y2>=LB(X2-XA) THEN 450           'W granicach: nie rysuj
420     LB(X2-XA) = Y2                      'Zmień granicę
430     IF UB(X2-XA)=0 THEN UB(X2-XA)=Y2    'Zmień także granicę górną
440     GOTO 470
450     IF Y2<=UB(X2-XA) THEN 480           'W granicach: nie rysuj
460     UB(X2-XA) = Y2                      'Zmień granicę
470     DRAWLINE X1,Y1 TO X2,Y2
480     X1 = X2                             'Zapamiętaj aktualny punkt
490     Y1 = Y2
500   NEXT X
510 NEXT Z
520 END
```



Rys. 9.20. Trójwymiarowa powierzchnia utworzona z linii z poprzedniego rysunku, z których usunięto niewidoczne fragmenty (program 9.9)

linii zrealizowano, kreśląc linie od przodu do tyłu i nie kreśląc odcinków niewidocznych, to znaczy tych, które pokrywałyby odcinki już wykreślone. Dla każdej wartości X zapamiętuje się największą i najmniejszą wartość Y z już narysowanych krzywych. Każda później obliczona krzywa, która znajdzie się w tych granicach, nie jest kreślona. Te części krzywej, które znajdują się poza uprzednio obliczonymi granicami, są kreślone i powodują odpowiednią zmianę tych granic. Dla oszczędności pamięci tablice UB i UL, w których zapamiętuje się granice, są zwymiarowane zgodnie z zakresem X. Zakres X może być obliczony z zależności określających X1 i X2 w wierszach 340 i 380 programu. Można tego dokonać analitycznie lub za pomocą krótkiego programu obliczającego wykreślone wartości X i drukującego maksymalną i minimalną wartość X. W naszym przykładzie zakres X1 wynosi 424. Parametrowi XA nadano minimalną wartość X1 − 1; używa się go jako przesunięcia przy odwzorowaniu kreślonej wartości X na indeks tablicy.

Korzystając z metod użytych w programie 9.9 można wykreślić dowolne zależności funkcjonalne między X, Y a Z. Wartości współrzędnych oblicza się z zależności analitycznych lub wprowadza jako dane tablicowe, na przykład jako dane topograficzne lub gęstości zaludnienia. Podobnie jak na rys. 9.18 można dodać osie i ich opisy.

ZADANIA

9.1. Wykonaj rysunek dowolnego obiektu trójwymiarowego na papierze milimetrowym, określając wartości (X, Y, Z) dla każdego punktu obiektu. Napisz program wyświetlenia rzutu prostokątnego tego obiektu.

9.2. Zmodyfikuj program 9.1 wyświetlenia dowolnego wielokąta. Ekran może być zapełniony trójkątami, prostokątami i innymi wielokątami. Granice każdej figury są liniami prostymi których nachylenia i przesunięcia wyznaczają punkty końcowe. Danymi wejściowymi do programu ma być zbiór wierzchołków każdego wielokąta.

9.3. Wykonaj na papierze milimetrowym rysunek dowolnej figury trójwymiarowej o ścianach w postaci wielokątów. Używając metod z programu 9.1 wyświetl tę figurę bez niewidocznych ścian rysując jej ściany kolejno od tyłu i stosując różne kolory.

9.4. Zmodyfikuj program 9.1 tak, aby wyświetlał okręgi zamiast prostokątów. Dla każdego okręgu podaj jego środek, promień i głębokość. Posortuj okręgi i narysuj je na ekranie, poczynając od okręgów o największej głębokości. Równanie granicy dolnej połówki okręgu jest następujące:

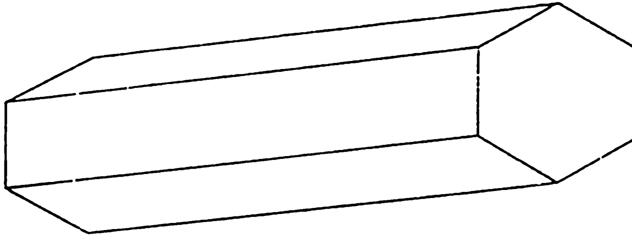
Y = YC + SQR(R^2 — (X — XC)^2)

a górnej

Y = YC — SQR(R^2 — (X — XC)^2)

Między tymi granicami narysuj linie pionowe przy wartości X zmieniającej się od XC − R do XC + R z krokiem jednostkowym.

9.5. Posłuż się metodą z programu 9.2 do usunięcia niewidocznych linii z rysunku belki o przekroju sześciokątnym (rys. 9.21). Wyświetl wszystkie linie belki, a następnie określ, która ściana z każdej z 4 par ścian przeciwległych jest niewidoczna i usuń ją.



Rys. 9.21. Belka o przekroju sześciokątnym

9.6. Używając metody z programu 9.4 usuń linie ukryte za okręgami. Program powinien wyświetlać dowolny okrąg i zadaną linię i usuwać tę część linii, która jest ukryta za okręgiem.

9.7. Wykonać na papierze milimetrowym rysunek obiektu trójwymiarowego, którego ściany są wielokątami wypukłymi. Korzystając z metody użytej w programie 9.4 napisz program usuwania niewidocznych linii. Skorzystaj następnie z równań perspektywy (9.1 dla dowolnie ustalonego punktu patrzenia (XV, YV, ZV)).

9.8. Zmodyfikuj program 9.6 tak, aby wyświetlał widoczne powierzchnie obiektów trójwymiarowych przy użyciu metody z programu 9.4 i zależności (9.2).

9.9. Zmodyfikuj program 9.4 tak, aby móc wprowadzać wielokąty wklęsłe i linie proste. Znajdź wszystkie punkty przecięcia (może być ich więcej niż dwa) linii z powierzchnią i usuń niewidoczne fragmenty.

9.10. Określ wzór cieniowania i użyj go w programie wyświetlania obiektów trójwymiarowych. Dla każdego obiektu zacieniuj ściany przeciwległe określonemu położeniu źródła światła.

9.11. Zmodyfikuj program 9.7 do wyświetlania trójwymiarowego wykresu słupkowego z podziałem każdego słupka na sekcje. Każdą sekcję pokoloruj inaczej.

9.12. Używając metody z programu 9.8 wykreśl powierzchnię trójwymiarową opisaną przez funkcję

```
Y = A * SIN(X) + B * SIN(3 * X)
```

Parametry A i B wybierz tak, aby rysunek mieścił się na ekranie dla zadanego zakresu X i Z.

10. Transformacje w trzech wymiarach

Rozszerzymy teraz transformacje na płaszczyźnie (przesunięcie, skalowanie i obrót), tak, aby włączyć trzeci wymiar — głębokość. Te dodatkowe metody zapewnią nam dużą elastyczność w manipulowaniu obrazami trójwymiarowymi i płaskimi.

10.1. Przesunięcie

Zależności określające przesunięcie na płaszczyźnie możemy rozszerzyć dopuszczając zmianę współrzędnej Z. Oznaczmy przez H wielkość przesunięcia w poziomie, przez V wielkość przesunięcia w pionie i przez D wielkość przesunięcia w głąb. Położenie punktu początkowego (X, Y, Z) po przesunięciu obliczamy jako:

$$\begin{aligned} XT &= X + H \\ YT &= Y + V \\ ZT &= Z + D \end{aligned} \qquad (10.1)$$

Dodatnie wartości H przesuwają punkt na prawo. Dodatnie wartości V przesuwają punkt w dół ekranu. Dodatnie wartości D przesuwają punkt dalej od nas, a ujemne wartości D bliżej nas. Wartości H, V i D nie powinny być zbyt duże, gdyż przesuwają wtedy punkty poza ekran, i nie powinny być za małe, gdyż nowe położenia punktów nie będą różnić się od poprzednich.

Rysunki i wykresy są przesuwane i animowane w przestrzeni, gdy takie same wartości przesunięcia H, V i D zastosujemy do wszystkich ich punktów. Jeśli użyjemy różnych wartości H, V i D do różnych punktów rysunku, to spowodujemy zmianę początkowego kształtu.

Przykład przesunięcia w trzech wymiarach realizuje program 10.1. Po wprowadzeniu do programu wartości przesunięć oblicza on nowe położenie punktów obiektu z równań perspektywy przedstawionych w rozdz. 9. Im większą wybieramy war-

Program 10.1. Przesunięcie w trzech wymiarach i rzut perspektywiczny (blok)

```
10  'PROGRAM 10.1. Przesunięcia w trzech wymiarach
20      'Program definiuje obiekt trójwymiarowy i przesuwa go
30      'w różne położenia. Obiekt jest wyświetlany w stosunku do
40      'ustalonego punktu patrzenia, położonego z przodu i nad środkiem
50      'ekranu. Widoczność ścian obiektu określa się na podstawie jego
60      'symetrii.
70  CLEARSCREEN
80  DIM X(8), Y(8), Z(8), XP(8), YP(8), D(8)
90  XM = 255
100 YM = 191
110     '******************** Odczytaj dane **********************
120 READ N
130 FOR K = 1 TO N
140   READ X(K), Y(K), Z(K)
150   IF X(K) < 0 OR X(K) > XM OR Y(K) < 0 OR Y(K) > YM THEN 1020
160 NEXT K
170 GRAPHICS
180 FOR K = 1 TO N/2 - 1
190   DRAWLINE X(K),Y(K) TO X(K+1),Y(K+1)
200 NEXT K
210 DRAWLINE X(N/2),Y(N/2) TO X(1),Y(1)
220 XV =28                   'Ustal punkt patrzenia nad środkiem ekranu
230 YV = 96
240 ZV = -100
250     '******************** Przesuń obiekt *********************
260 PRINT "Wprowadź wielkości przesunięć H, V i D"
270 PRINT "Napisz 1000, 1000, 1000 aby zakończyć"
280 INPUT H, V, D
290 IF H = 1000 AND V = 1000 AND D = 1000 THEN 1020
300     'Oblicz nowe punkty
310 FOR K =1 TO N
320   XT(K) = X(K) + H
330   YT(K) = Y(K) + V
340   ZT(K) = Z(K) + D
350 NEXT K
360     'Oblicz odległości i określ perspektywę punktów
370 FOR K = 1 TO N
380   D(K) = SQR((XT(K)-XV)^2 + (YT(K)-YV)^2 +(ZT(K)-ZV)^2)
390   XP(K) = XV + (XT(K) - XV) * (-ZV) / (ZT(K) - ZV)
400   YP(K) = YV + (YT(K) - YV) * (-ZV) / (ZT(K) - ZV)
410   IF XP(K)>=0 AND XP(K)<=XM AND YP(K)>=0 AND YP(K)<=YM THEN 440
420   PRINT "Współrzędne poza ekranem. Spróbuj jeszcze raz"
430   GOTO 260
440 NEXT K
450 GOSUB 480                  'Rysuj
460 GOTO 260
470 '
480 '****************** Program rysowania ************************
490 CLEARSCREEN
500     '************* Sciana przednia i tylna ***************
510 IF D(1) = D(5) THEN 670          'Żadna ściana nie jest widoczna
520 IF D(1) > D(5) THEN 600
530 IF ZT(1) <= ZV THEN 670          'Czy ściana jest widoczna?
540 FOR K = 1 TO 3                   'Rysuj ścianę zawierającą punkt 1
550   DRAWLINE XP(K),YP(K) TO XP(K+1),YP(K+1)
560 NEXT K
570 DRAWLINE XP(4),YP(4) TO XP(1),YP(1)
580 GOTO 670
590     '
600 IF ZT(5) >= ZV THEN 670          'Czy ściana jest widoczna?
610 FOR K = 5 TO 7                   'Rysuj ścianę zawierającą punkt 5
620   DRAWLINE XP(K),YP(K) TO XP(K+1),YP(K+1)
630 NEXT K
```

Program 10.1 (cd.)

```
640 DRAWLINE XP(8),YP(8) TO XP(5),YP(5)
650     '
660     '****************** Sciany boczne *******************
670 IF D(1) = D(4) THEN 830          'Żadna ściana nie jest widoczna
680 IF D(1) > D(4) THEN 760
690 IF XT(1) >= XV THEN 830                 'Czy ściana jest widoczna?
700 DRAWLINE XP(1),YP(1) TO XP(2),YP(2)    'Rysuj ścianę zaw. punkt 1
710 DRAWLINE XP(2),YP(2) TO XP(6),YP(6)
720 DRAWLINE XP(6),YP(6) TO XP(5),YP(5)
730 DRAWLINE XP(5),YP(5) TO XP(1),YP(1)
740 GOTO 830
750     '
760 IF XT(4) <= XV THEN 830                 'Czy ściana jest widoczna?
770 DRAWLINE XP(4),YP(4) TO XP(3),YP(3)    'Rysuj ścianę zaw. punkt 4
780 DRAWLINE XP(3),YP(3) TO XP(7),YP(7)
790 DRAWLINE XP(7),YP(7) TO XP(8),YP(8)
800 DRAWLINE XP(8),YP(8) TO XP(4),YP(4)
810     '
820     '**************** Sciany górna i dolna ***************
830 IF D(1) = D(2) THEN 970          'Żadna ściana nie jest widoczna
840 IF D(1) > D(2) THEN 920
850 IF YT(1) <= YV THEN 970                 'Czy ściana jest widoczna?
860 DRAWLINE XP(1),YP(1) TO XP(4),YP(4)    'Rysuj ścianę zaw. punkt 1
870 DRAWLINE XP(4),YP(4) TO XP(8),YP(8)
880 DRAWLINE XP(8),YP(8) TO XP(5),YP(5)
890 DRAWLINE XP(5),YP(5) TO XP(1),YP(1)
900 GOTO 970
910     '
920 IF YT(2) >= YV THEN 970
930 DRAWLINE XP(2),YP(2) TO XP(3),YP(3)    'Rysuj ścianę zaw. punkt 2
940 DRAWLINE XP(3),YP(3) TO XP(7),YP(7)
950 DRAWLINE XP(7),YP(7) TO XP(6),YP(6)
960 DRAWLINE XP(6),YP(6) TO XP(2),YP(2)
970 RETURN
980     '************************************************************
990  DATA 8
1000 DATA 138,56,0,138,136,0,118,136,0,118,56,0
1010 DATA 138,56,40,138,136,40,118,136,40,118,56,40
1020 END
```

Wprowadź wielkości przesunięć H, V i D
Napisz 1000, 1000, 1000 aby zakończyć



(a)

```
Wprowadź wielkości przesunięć H, V i D
Napisz 1000, 1000, 1000 aby zakończyć
```



(b)

Rys. 10.1. Wyniki programu 10.1: obiekt oryginalny (a) i przesunięty widok obiektu (b) widziany z punktu patrzenia o współrzędnych (128, 96, −100)

tość D, tym mniejszy będzie wymiar obiektu i tym bliżej punktu zbiegu będzie położony. Na rysunku 10.1 pokazano obiekt oryginalny i przesunięty o H = 60, V = 60 i D = 20.

10.2. Skalowanie

Uogólnienie zależności skalowania w dwóch wymiarach na trzy wymiary odbywa się przez powiększanie lub zmniejszanie wzdłuż osi Z — oprócz osi X i Y. Podobnie jak w przypadku dwuwymiarowym wybieramy stały punkt i obliczamy nowe położenia wszystkich punktów w stosunku do tego stałego punktu używając zadanych współczynników skali. Przyjmując, że współrzędne punktu stałego wynoszą (XF, YF, ZF), obliczamy współrzędne (XS, YS, ZS) po skalowaniu punktu o współrzędnych początkowych (X, Y, Z) jako:

$$
\begin{aligned}
XS &= X * HS + XF * (1 - HS) \\
YS &= Y * VS + YF * (1 - VS) \\
ZS &= T * DS + ZF * (1 - DS)
\end{aligned}
\qquad (10.2)
$$

W zależności tej współczynniki skali wzdłuż każdej osi oznaczono jako HS, VS i DS.

Każdy z tych współczynników może przyjąć dowolną wartość dodatnią. Wartości większe niż 1 powodują powiększanie rysunku, a wartości mniejsze niż 1 zmniejszanie.

Zwykle, współczynnikom HS, VS i DS nadaje się takie same wartości. Powoduje to powiększanie lub zmniejszanie obiektu równomiernie we wszystkich kierunkach. Jeśli chcemy powiększyć lub zmniejszyć obiekt w różny sposób w różnych kierunkach, to stosujemy różne współczynniki skali. Duże wartości współczynników skali mogą przesunąć obraz poza ekran. Małe wartości współczynników skali mogą zredukować go do wymiaru pojedynczego punktu.

Skalowanie obiektu zgodnie z zależnościami (10.2) zilustrowano w programie 10.2. Program ten skaluje figurę robota do dowolnych wymiarów w stosunku do punktu stałego znajdującego się na robocie. Początkowo robot umieszczono w lewym dolnym rogu ekranu, przodem do obserwatora. Położenie obserwatora wybrano z prawej strony nad robotem. Widoczność ścian robota jest określana względem tego punktu, a skalowany robot jest wyświetlany od dołu i od lewej do prawej strony. Na rysunku 10.2 pokazano w perspektywie wynik zmniejszania i zwiększania wymiarów robota przy jednakowych współczynnikach skali we wszystkich kierunkach — uzyskano równomierne skalowanie. Można tak przewidzieć dane wejściowe do programu, aby umożliwić różne definicje obiektu, zmiany punktu stałego i punktu patrzenia oraz dowolne współczynniki skali HS, VS i DS.

Program 10.2. Skalowanie w trzech wymiarach i rzut perspektywiczny (robot)

```
10 'PROGRAM 10.2. Skalowanie w trzech wymiarach
20      'Program definiuje obraz robota w trzech wymiarach i skaluje
30      'go w stosunku do punktu (XF,YF,ZF). Punkt patrzenia ma współ-
40      'rzędne XV,YV,ZV. Korzystając z częściowej symetrii robota
50      'określa się, która ściana jest widoczna. Wnętrze każdej
60      'widocznej ściany jest wymazane, co usuwa wszystkie linie
70      'niewidoczne.
80  CLEARSCREEN
90  DIM X(10,4), Y(10,4), Z(10,4), XS(10,4), YS(10,4), ZS(10,4)
100 DIM XP(10,4), YP(10,4), D(10,4)
110 XM = 319
120 YM = 199               'XM i YM są maksymalnymi wartościami dla systemu
130 XF = 80                'Rysunek skaluje się względem punktu (XF,YF,ZF)
140 YF = 110
150 ZF = 7
160 XV = 160               'XV,YV,ZV są współrzędnymi punktu patrzenia
170 YV = 10
180 ZV = -100
190     '*************** Odczytaj dane do rysunku *************
200 READ NS                'NS jest liczbą ścian
210 FOR S = 1 TO NS
220   READ NV(S)           'NV jest liczbą wierzchołków ściany
230   FOR P = 1 TO NV(S)
240     READ X(S,P), Y(S,P), Z(S,P)
250     IF X(S,P)<0 OR X(S,P)>XM OR Y(S,P)<0 OR Y(S,P)>YM THEN 1800
260   NEXT P
270 NEXT S
280 GRAPHICS
290 HS = 1                 'Początkowa wartość współczynników skali równa 1
300 VS = 1
310 DS = 1
```

Program 10.2 (cd.)

```
320 GOTO 380
330     '*************** Przeskaluj obiekt ********************
340 PRINT "Wprowadź współczynniki skali HS, VS i DS"
350 PRINT "Napisz 0,0,0 aby zakończyć program"
360 INPUT HS, VS, DS
370 IF HS = 0 AND VS = 0 AND DS = 0 THEN 1800
380     '******* Oblicz nowe wartości współrzędnych punktów *******
390 FOR S = 1 TO NS
400   FOR P = 1 TO NV(S)
410     XS(S,P) = X(S,P) * HS + XF * (1 - HS)
420     YS(S,P) = Y(S,P) * VS + YF * (1 - VS)
430     ZS(S,P) = Z(S,P) * DS + ZF * (1 - DS)
440     D(S,P) = SQR((XS(S,P)-XV)^2 + (YS(S,P)-YV)^2 + (ZS(S,P)-ZV)^2)
450     XP(S,P) = XV + (XS(S,P) - XV) * (-ZV) / (ZS(S,P) - ZV)
460     YP(S,P) = YV + (YS(S,P) - YV) * (-ZV) / (ZS(S,P) - ZV)
470     IF XP(S,P)>=0 AND XP(S,P)<=XM AND
           YP(S,P)>=0 AND YS(S,P)<=YM THEN 500
480     PRINT "Poza ekranem. Spróbuj jeszcze raz"
490     GOTO 330
500   NEXT P
510 NEXT S
520 GOSUB 540                    'Rysuj figurę
530 GOTO 330
540 '****************** Program rysowania *********************
550 CLEARSCREEN
560 FOR S = 1 TO 8 STEP 2        'Sprawdź każdą parę symetrycznych ścian
570   '
580   '************** Sciany przednia i tylna ******************
590   IF D(S,1) = D(S+1,1) THEN 910     'Żadna ściana nie jest widoczna
600   IF D(S,1) > D(S+1,1) THEN 800
610   IF ZS(S,1) <= ZV THEN 910         'Czy ściana jest widoczna?
620   'Wymaż wnętrze ściany przedniej
630   FOR X = XP(S,1) TO XP(S,4) STEP -1
640     DRAWLINE X,YP(S,1) TO X,YP(S,2)
650   NEXT X
660   'Rysuj obrys ściany przedniej (ściany zawierającej punkt (S,1))
670   FOR K = 1 TO 3
680     DRAWLINE XP(S,K),YP(S,K) TO XP(S,K+1), YP(S,K+1)
690   NEXT K
700   DRAWLINE XP(S,4),YP(S,4) TO XP(S,1),YP(S,1)
710   IF S <> 7 THEN 910                'Narysuj oczy robota
720   FOR K = 1 TO 3
730     DRAWLINE XP(9,K),YP(9,K) TO XP(9,K+1),YP(9,K+1)
740     DRAWLINE XP(10,K),YP(10,K) TO XP(10,K+1),YP(10,K+1)
750   NEXT K
760   DRAWLINE XP(9,4),YP(9,4) TO XP(9,1),YP(9,1)
770   DRAWLINE XP(10,4),YP(10,4) TO XP(10,1),YP(10,1)
780   GOTO 910
800   IF ZS(S+1,1) >= ZV THEN 910       'Czy ściana jest widoczna?
810   'Wymaż wnętrze ściany tylnej
820   FOR X = XP(S+1,1) TO XP(S+1,4) STEP -1
830     DRAWLINE X,YP(S+1,1) TO X,YP(S+1,2)
840   NEXT X
850   'Rysuj obrys ściany tylnej (ściany zawierającej punkt (S+1,1))
860   FOR K = 1 TO 3
870     DRAWLINE XP(S+1,K),YP(S+1,K) TO XP(S+1,K+1),YP(S+1,K+1)
880   NEXT K
890   DRAWLINE XP(S+1,4),YP(S+1,4) TO XP(S+1,1),YP(S+1,1)
900   '
910   '***************** Sciany boczne *********************
920   IF D(S,1) = D(S,4) THEN 1290      'Żadna ściana nie jest widoczna
930   IF D(S,1) > D(S,4) THEN 1120
940   IF XS(S,1) >= XV THEN 1290        'Czy ściana jest widoczna?
950   'Wymaż wnętrze ściany prawej
```

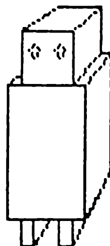
Program 10.2 (cd.)

```
960   MT = (YP(S,1) - YP(S+1,1)) / (XP(S,1) - XP(S+1,1))
970   BT = YP(S,1) - (MT * XP(S,1))
980   MB = (YP(S,2) - YP(S+1,2)) / (XP(S,2) - XP(S+1,2))
990   BB = YP(S,2) - (MP * XP(S,2))
1000   FOR X = XP(S,1) TO XP(S+1,1)
1010     YT = MT * X + BT
1020     YB = MB * X + BB
1030     DRAWLINE X,YT TO X,YB
1040   NEXT X
1050   'Rysuj obrys ściany prawej (ściany zawierającej punkt (S,1))
1060   DRAWLINE XP(S,1),YP(S,1) TO XP(S,2),YP(S,2)
1070   DRAWLINE XP(S,2),YP(S,2) TO XP(S+1,2),YP(S+1,2)
1080   DRAWLINE XP(S+1,2),YP(S+1,2) TO XP(S+1,1),YP(S+1,1)
1090   DRAWLINE XP(S+1,1),YP(S+1,1) TO XP(S,1),YP(S,1)
1100   GOTO 1290
1110   '
1120   IF XS(S,4) <= XV THEN 1290         'Czy ściana jest widoczna?
1130   'Wymaż wnętrze ściany lewej
1140   MT = (YP(S,4) - YP(S+1,4)) / (XP(S,4) - XP(S+1,4))
1150   BT = YP(S,4) - MT * XP(S,4)
1160   MB = (YP(S,3) - YP(S+1,3)) / (XP(S,3) - XP(S+1,3))
1170   BB = YP(S,3) - MB * XP(S,3)
1180   FOR X = XPS(,4) TO XPS(+1,4) STEP -1
1190     YT = MT * X + BT
1200     YB = MB * X + BB
1210     DRAWLINE X,YT TO X,YB
1220   NEXT X
1230   'Rysuj obrys ściany lewej (ściany zawierającej punkt (S,4))
1240   DRAWLINE XP(S,4),YP(S,4) TO XP(S,3),YP(S,3)
1250   DRAWLINE XP(S,3),YP(S,3) TO XP(S+1,3),YP(S+1,3)
1260   DRAWLINE XP(S+1,3),YP(S+1,3) TO XP(S+1,4),YP(S+1,4)
1270   DRAWLINE XP(S+1,4),YP(S+1,4) TO XP(S,4),YP(S,4)
1280   '
1290   '*************** Sciany górna  i dolna ********************
1300   IF D(S,1) = D(S,2) THEN 1660       'Żadna ściana nie jest widoczna
1310   IF D(S,1) > D(S,2) THEN 1500
1320   IF YS(S,1) <= YV THEN 1660         'Czy ściana jest widoczna?
1330   'Wymaż wnętrze ściany górnej
1340   ML = (YP(S+1,4) - YP(S,4)) / (XP(S+1,4) - XP(S,4))
1350   BL = YP(S,4) - ML * XP(S,4)
1360   MR = (YP(S+1,1) - YP(S,1)) / (XP(S+1,1) - XP(S,1))
1370   BR = YP(S,1) - MP * XP(S,1)
1380   FOR Y = YP(S+1,4) TO YP(S,4)
1390     XL = (Y - BL) / ML
1400     XR = (Y - BR) / MR
1410     DRAWLINE XL,Y TO XR,Y
1420   NEXT Y
1430   'Rysuj obrys ściany górnej (ściany zawierającej punkt (S,1))
1440   DRAWLINE XP(S,1),YP(S,1) TO XP(S,4),YP(S,4)
1450   DRAWLINE XP(S,4),YP(S,4) TO XP(S+1,4),YP(S+1,4)
1460   DRAWLINE XP(S+1,4),YP(S+1,4) TO XP(S+1,1),YP(S+1,1)
1470   DRAWLINE XP(S+1,1),YP(S+1,1) TO XP(S,1),YP(S,1)
1480   GOTO 1660
1490   '
1500   IF XS(S,2) >= YV THEN 1660         'Czy ściana jest widoczna?
1510   'Wymaż wnętrze ściany dolnej
1520   ML = (YP(S+1,3) - YP(S,3)) / (XP(S+1,3) - XP(S,3))
1530   BL = YP(S,3) - ML * XP(S,3)
1540   MR = (YP(S+1,2) - YP(S,2)) / (XP(S+1,2) - XP(S,2))
1550   BR = YP(S,2) - MR * XP(S,2)
1560   FOR Y = YP(S+1,3) TO YP(S,3) STEP -1
1570     XL = (Y - BL) / ML
1580     XR = (Y - BR) /MR
1590     DRAWLINE XL,Y TO XR,Y
```

Program 10.2 (cd.)

```
1600    NEXT Y
1610    'Rysuj obrys ściany dolnej (ściany zawierającej punkt (S,2))
1620    DRAWLINE XP(S,2),YP(S,2) TO XP(S,3),YP(S,3)
1630    DRAWLINE XP(S,3),YP(S,3) TO XP(S+1,3),YP(S+1,3)
1640    DRAWLINE XP(S+1,3),YP(S+1,3) TO XP(S+1,2),YP(S+1,2)
1650    DRAWLINE XP(S+1,2),YP(S+1,2) TO XP(S,2),YP(S,2)
1660 NEXT S
1670 RETURN
1680    '************************************************************
1690 DATA 10
1700 DATA 4,60,140,0,60,150,0,55,150,0,55,140,0
1710 DATA 4,60,140,15,60,150,15,55,150,15,55,140,15
1720 DATA 4,75,140,0,75,150,0,70,150,0,70,140,0
1730 DATA 4,75,140,15,75,150,15,70,150,15,70,140,15
1740 DATA 4,80,90,0,80,140,0,50,140,0,50,90,0
1750 DATA 4,80,90,15,80,140,15,50,140,15,50,90,15
1760 DATA 4,75,70,0,75,90,0,55,90,0,55,70,0
1770 DATA 4,75,70,15,75,90,15,55,90,15,55,70,15
1780 DATA 4,70,75,0,72,77,0,70,80,0,68,77,0
1790 DATA 4,60,75,0,62,77,0,60,80,0,58,77,0
1800 END
```

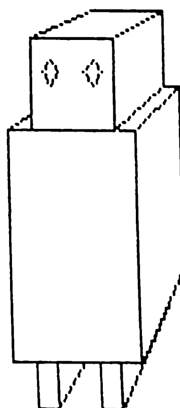
Wprowadź współczynniki skali HS, VS i DS
Napisz 0, 0, 0 aby zakończyć program



(a)

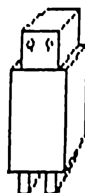
Rys. 10.2a (podpis na stronie 224)

```
Wprowadź współczynniki skali HS, VS i DS
Napisz 0, 0, 0 aby zakończyć program
```



(b)

```
Wprowadź współczynniki skali HS, VS i DS
Napisz 0, 0, 0 aby zakończyć program
```



(c)

Rys. 10.2. Skalowanie obiektu trójwymiarowego za pomocą programu 10.2: oryginał w rzucie perspektywicznym (a), obiekt powiększony (b) obiekt zmniejszony (c). Punkt patrzenia ma współrzędne (160, 10, −100)

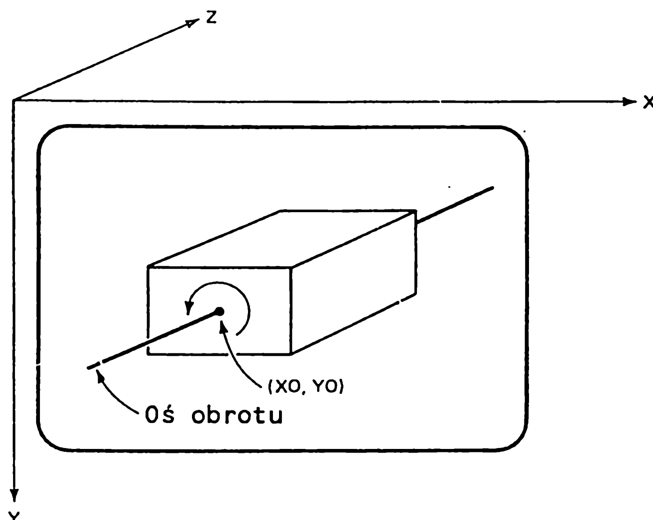
Te same techniki skalowania stosuje się do wykresów trójwymiarowych Jednakże w wielu przypadkach można przedstawić zależności między trzema lub więcej zmiennymi tylko w dwóch wymiarach, we współrzędnych ekranu. W takim przypadku do skalowania rysunku wystarczą metody omówione w rozdz. 6.

10.3. Obrót

Widzieliśmy już, że obiekt może być obrócony o kąt A wokół ustalonego punktu (X0, Y0) w płaszczyźnie X, Y przez transformację wszystkich współrzędnych X i Y we współrzędne XR, YR według zależności:

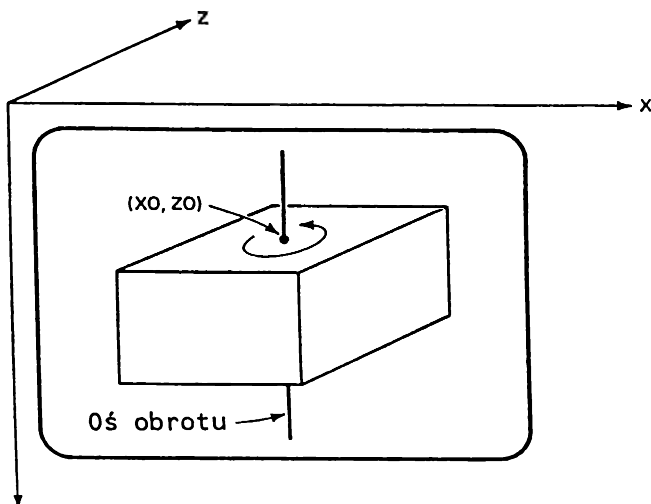
$$\begin{aligned} XR &= X0 + (X - X0) * COS(A) + (Y - Y0) * SIN(A) \\ YR &= Y0 + (Y - Y0) * COS(A) - (X - X0) * SIN(A) \end{aligned} \qquad (10.3)$$

W tych zależnościach kąt podawany w radianach jest mierzony w przeciwnym kierunku względem ruchu wskazówek zegara od punktu (X, Y) do punktu (XR, YR).



Rys. 10.3. Obrót obiektu wokół osi równoległej do osi Z zmienia wartości współrzędnych X i Y, nie zmienia wartości współrzędnej Z

Rozważmy obiekt trójwymiarowy pokazany na rys. 10.3. Załóżmy, że obrót następuje wokół linii przechodzącej przez punkt (X0, Y0), równoległej do osi Z. Wszystkie punkty obiektu będą obrócone wokół tej linii. Taki obrót nie zmieni wartości współrzędnej Z.



Rys. 10.4. Obrót obiektu wokół osi równoległej do osi Y zmienia wartości współrzędnych X i Z, nie zmienia wartości współrzędnej Y

W podobny sposób możemy rozważać obrót obiektów trójwymiarowych wokół osi w innych kierunkach. Na rysunku 10.4 pokazano oś obrotu równoległą do osi Y, przechodzącą przez punkt (X0, Z0). Obrót wokół tej osi zmieni współrzędne X i Z punktów na nowe współrzędne XR i ZR, które możemy obliczyć z zależności:

$$\begin{aligned} XR &= X0 + (X - X0) * COS(A) - (Z - Z0) * SIN(A) \\ ZR &= Z0 + (Z - Z0) * COS(A) + (X - X0) * SIN(A) \end{aligned} \tag{10.4}$$

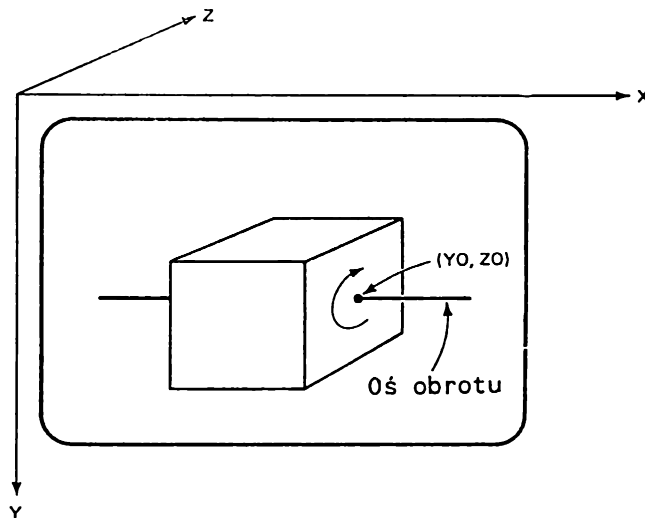
Taki obrót nie zmienia wartości Y. Kąt w tych zależnościach liczy się w kierunku przeciwnym względem ruchu wskazówek zegara, gdy patrzymy z „góry" pudełka. Inny sposób opisania tego obrotu polega na wyobrażeniu sobie, że stoimy w początku układu współrzędnych i patrzymy wzdłuż dodatniej połówki osi Y, a obiekt obraca się w kierunku przeciwnym względem ruchu wskazówek zegara. Jeśli obiekt ma obracać się w przeciwną stronę, to wystarczy przyjąć ujemne wartości kąta obrotu A (w radianach) w zależnościach (10.4).

Na rysunku 10.5 oś obrotu jest równoległa do osi X. Przy tym obrocie każda ze współrzędnych Y i Z punktów zmieni się na współrzędne YR i ZR. Nowe wartości współrzędnych oblicza się z zależności:

$$\begin{aligned} XR &= Y0 + (Y - Y0) * COS(A) + (Z - Z0) * SIN(A) \\ ZR &= Z0 + (Z - Z0) * COS(A) - (Y - Y0) * SIN(A) \end{aligned} \tag{10.5}$$

Taki obrót nie zmienia wartości X. Gdy patrzymy na pudełko z „lewej" strony (albo od początku układu współrzędnych wzdłuż dodatniej połówki osi X), dodatnie wartości kąta A odpowiadają obrotowi w kierunku przeciwnym względem ruchu wskazówek zegara.

Można wybrać także inne osie obrotu, ale dowolny obrót daje się złożyć z trzech przedstawionych obrotów. Obracanie obiektu wokół osi równoległej do osi Z (rys. 10.3) powoduje, że stale oglądamy tę samą stronę. Obracanie obiektu wokół osi równoległej do osi Y (rys. 10.4) pokazuje nam kolejno różne ściany obiektu. Obracanie obiektu wokół osi równoległej do osi X (rys. 10.5) umożliwia obejrzenie góry i dołu obiektu. W celu połączenia różnych widoków obiektu, na przykład z góry i z lewej strony, musimy dokonać dwóch lub więcej obrotów obiektu. Kolejność obrotów jest ważna, gdy zmieniając ją możemy otrzymać zupełnie inny widok obiektu.



Rys. 10.5. Obrót obiektu wokół osi równoległej do osi X zmienia wartości współrzędnych Y i Z nie zmienia wartości współrzędnej X

Program 10.3 umożliwia obrót kostki do gry w dowolnej kombinacji trzech obrotów podstawowych. Na rysunku 10.6 pokazano wyniki obrotu kostki wokół trzech osi. W każdym z przypadków oś obrotu przechodzi przez środek kostki. Przy obracaniu obiektów nie stosowano rzutu perspektywicznego.

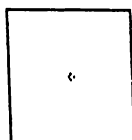
Program 10.3. Obroty w trzech wymiarach (kostka do gry)

```
10  'PROGRAM 10.3. Obrót w trzech wymiarach
20      'Program definiuje w trzech wymiarach pojedynczą kostkę do gry.
30      'Dane wierzchołków i wszystkie oznakowania są zapamiętane w
40      'tablicach XR, YR, ZR. Kostka może być obrócona o dowolny kąt
50      'wokół dowolnej z osi. Do określenia, które ściany są widoczne,
60      'korzysta się z symetrii kostki.
70      '************************************************************
80  CLEARSCREEN
90  DIM XR(29), YR(29), ZR(29)
100 XM = 319
110 YM = 199
120 READ XO, YO, ZO                      'Obrót wokół tego punktu
130 FOR K = 1 TO 29
140   READ XR(K), YR(K), ZR(K)
150   IF XR(K)<O OR XR(K)>XM OR YR(K)<O OR YR(K)>YM THEN 1150
160 NEXT K
170 GRAPHICS
180 GOSUB 510
190 POSITION 1,1
200 PRINT "Napisz K aby zakończyć program"
210 PRINT "Wokół której osi obrócić?"
220 INPUT R$
230 IF R$ = "K" THEN 1150
240 PRINT "O ile stopni?"
250 INPUT A
260 A = A * 3.14159 / 180               'Zamiana na radiany
270     '*********** Oblicz nowe współrzędne punktów ***************
280 IF R$ = "X" THEN 310
290 IF R$ = "Y" THEN 370
300 IF R$ = "Z" THEN 430
310 FOR K = 1 TO 29                     'Obrót wokół osi X
320   YS = YR(K)           'Zapamiętaj YR(K) do obliczania ZR
330   YR(K) = INT(YO + (YR(K) - YO)*COS(A) + (ZR(K) - ZO)*SIN(A) + 0.5)
340   ZR(K) = INT(ZO + (ZR(K) - ZO)*COS(A) - (YS - YO)*SIN(A) + 0.5)
350 NEXT K
360 GOTO 480
370 FOR K = 1 TO 29                     'Obrót wokół osi Y
380   XS = XR(K)           'Zapamiętaj XR(K) do obliczania ZR
390   XR(K) = INT(XO + (XR(K) - XO)*COS(A) - (ZR(K) - ZO)*SIN(A) + 0.5)
400   ZR(K) = INT(ZO + (ZR(K) - ZO)*COS(A) + (XS - XO)*SIN(A) + 0.5)
410 NEXT K
420 GOTO 480
430 FOR K = 1 TO 29                     'Obrót wokół osi Z
440   XS = XR(K)           'Zapamiętaj XR(K) do obliczania YR
450   XR(K) = INT(XO + (XR(K) - XO)*COS(A) + (YR(K) - YO)*SIN(A) + 0.5)
460   YR(K) = INT(YO + (YR(K) - YO)*COS(A) - (XS - XO)*SIN(A) + 0.5)
470 NEXT K
480 GOSUB 500
490 GOTO 190
500     '************* Rysuj jedynie widoczne ściany ****************
510 CLEARSCREEN
520 IF ZR(1) = ZR(5) THEN 680         'Żadna ściana nie jest widoczna
530 IF ZR(1) > ZR(5) THEN 610
540 FOR K = 1 TO 3                    'Rysuj ścianę z 1 kropką
550   DRAWLINE XR(K),YR(K) TO XR(K+1),YR(K+1)
560 NEXT K
570 DRAWLINE XR(4),YR(4) TO XR(1),YR(1)
580 CIRCLEPLOT XR(9),YR(9),1
590 GOTO 680
600     '
```

Program 10.3 (cd.)

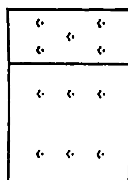
```
610 FOR K = 5 TO 7                        'Rysuj ścianę z 6 kropkami
620   DRAWLINE XR(K),YR(K) TO XR(K+1),YR(K+1)
630 NEXT K
640 DRAWLINE XR(8),YR(8) TO XR(5),YR(5)
650 FOR K = 24 TO 29
660   CIRCLEPLOT XR(K),YR(K),1
670 NEXT K
680 IF ZR(1) = ZR(4) THEN 860             'Zadna ściana nie jest widoczna
690 IF ZR(1) > ZR(4) THEN 790
700 DRAWLINE XR(1),YR(1) TO XR(2),YR(2)   'Rysuj ścianę z 4 kropkami
710 DRAWLINE XR(2),YR(2) TO XR(6),YR(6)
720 DRAWLINE XR(6),YR(6) TO XR(5),YR(5)
730 DRAWLINE XR(5),YR(5) TO XR(1),YR(1)
740 FOR K = 15 TO 18
750   CIRCLEPLOT XR(K),YR(K),1
760 NEXT K
770 GOTO 860
780   '
790 DRAWLINE XR(4),YR(4) TO XR(3),YR(3)   'Rysuj ścianę z 3 kropkami
800 DRAWLINE XR(3),YR(3) TO XR(7),YR(7)
810 DRAWLINE XR(7),YR(7) TO XR(8),YR(8)
820 DRAWLINE XR(8),YR(8) TO XR(4),YR(4)
830 FOR K = 12 TO 14
840   CIRCLEPLOT XR(K),YR(K),1
850 NEXT K
860 IF ZR(1) = ZR(2) THEN 1040            'Zadna ściana nie jest widoczna
870 IF ZR(1) > ZR(2) THEN 970
880 DRAWLINE XR(1),YR(1) TO XR(4),YR(4)   'Rysuj ścianę z 2 kropkami
890 DRAWLINE XR(4),YR(4) TO XR(8),YR(8)
900 DRAWLINE XR(8),YR(8) TO XR(5),YR(5)
910 DRAWLINE XR(5),YR(5) TO XR(1),YR(1)
920 FOR K = 10 TO 11
930   CIRCLEPLOT XR(K),YR(K),1
940 NEXT K
950 GOTO 1040
960   '
970 DRAWLINE XR(2),YR(2) TO XR(3),YR(3)   'Rysuj ścianę z 5 kropkami
980 DRAWLINE XR(3),YR(3) TO XR(7),YR(7)
990 DRAWLINE XR(7),YR(7) TO XR(6),YR(6)
1000 DRAWLINE XR(6),YR(6) TO XR(2),YR(2)
1010 FOR K = 19 TO 23
1020   CIRCLEPLOT XR(K),YR(K),1
1030 NEXT K
1040 RETURN
1050   '****************************************************************
1060 DATA 140,80,124
1070 DATA 164,56,100,164,104,100,116,104,100,116,56,100
1080 DATA 164,56,148,164,104,148,116,104,148,116,56,148
1090 DATA 140,80,100
1100 DATA 128,56,136,152,56,112
1110 DATA 116,68,136,116,80,124,116,92,112
1120 DATA 164,68,112,164,68,136,164,92,112,164,92,136
1130 DATA 128,104,112,128,104,136,140,104,124,152,104,112,152,104,136
1140 DATA 128,68,148,140,68,148,152,68,148,128,92,148,140,92,148,152,92,148
1150 END
```

```
Napisz K aby zakończyć program
Wokół której osi obrócić?
```



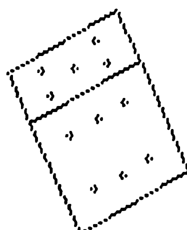
(a)

```
Napisz K aby zakończyć program
Wokół której osi obrócić?
```



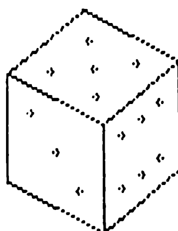
(b)

```
Napisz K aby zakończyć program
Wokół której osi obrócić?
```



(c)

```
Napisz K aby zakończyć program
Wokół której osi obrócić?
```



(d)

Rys. 10.6. Kostka do gry obracana wokół różnych osi (program 10.3): a) widok oryginalny; b) obrót o 155° wokół osi X; c) obrót o 28° wokół osi Z; d) obrót o 45° wokół osi Y

10.4. Łączenie transformacji

Programy różnych transformacji obiektów można połączyć w jeden zestaw programów. Taki zestaw organizuje się tak, że wprowadzony do programu obiekt jest wyświetlany i transformowany przez dowolną kombinację przesunięcia, skalowania, obrotów i rzutów perspektywicznych. Do wyboru transformacji możemy stosować procedurę wyboru konwersacyjnego z ustalonym sygnałem zakończenia. Do wyświetlenia jedynie widocznych ścian obiektu można korzystać z procedur usuwania niewidocznych linii lub niewidocznych ścian.

ZADANIA

10.1. Dokonaj animacji obiektu w trzech wymiarach. Obiekt powinien być przesuwany do przodu i do tyłu po ekranie ze stałymi dodatnimi i ujemnymi wartościami przyrostów X i Z. Najpierw przesuń obiekt z lewej na prawą stronę i zwiększ wartość Z. Gdy obiekt osiągnie prawy brzeg ekranu, zmień kierunek ruchu wzdłuż osi X pozostawiając wartości współrzędnych Y i Z nie zmienione. Gdy obiekt powróci na lewą stronę ekranu, zmień kierunek ruchu wzdłuż osi X i zacznij zwiększać wartość współrzędnej Z w kierunku ujemnym. Powtarzając ten ruch w perspektywie, doprowadź do ruchów obiektu wzdłuż figury o kształcie ósemki w płaszczyźnie X, Z.

10.2. Zmodyfikuj program z poprzedniego zadania tak, aby można było poruszać obiekt wokół dowolnej krzywej w trzech wymiarach.

10.3. Dokonaj animacji obiektu wzdłuż drogi złożonej z odcinków linii prostych w trzech wymiarach, biegnącej za prostokątną ścianę narysowaną na ekranie. Nie wyświetlaj obiektu, gdy znajduje się za ścianą. Program można uogólnić na przypadek wielu „ścian" o różnych kształtach.

10.4. Napisz program skalowania dowolnego obiektu dwu- lub trójwymiarowego zwiększając jego współrzędną Z i stosując równania perspektywy.

10.5. Opracuj program skalowania obiektu trójwymiarowego w dowolnym kierunku. Dokonaj tego obracając obiekt do określonego położenia, a następnie skalując wzdłuż osi X, Y i Z.

10.6. Napisz program wyświetlający napis dużymi literami, które maleją ze wzrostem odległości. Można tego dokonać obracając słowa wzdłuż osi pionowej (równoległej do osi Y) przechodzącej z lewej strony słowa i korzystając z równań perspektywy. Słowa te można następnie przesunąć do dowolnej pozycji na ekranie.

10.7. Napisz program wyświetlania kuli ziemskiej tworzonej przez obracający się okrąg. Obracając okrąg o 30° w trzech krokach wokół średnicy równoległej do osi Y uzyskuje się kulę z liniami południków. Przyjmując mniejsze kroki obrotu okręgu w zakresie od 0 do 90° otrzymujemy więcej linii południków. Linie równoleżników rysujemy poziomo między granicami początkowego okręgu.

10.8. Powtórz metodę z zadania 10.7 w celu utworzenia cylindra przez obracanie prostokąta wokół osi pionowej, a następnie wokół osi poziomej w celu pokazania jego góry lub dołu. Usuń niewidoczne linie.

10.9. Napisz program wyświetlenia figury trójwymiarowej obrotowej utworzonej z odcinka linii prostej. Narysuj ukośną linię z różnymi wartościami współrzędnej Z dla jej końców. Następnie obróć tę linię wokół osi pionowej (równoległej do osi Y) przechodzącej przez środek odcinka. Obracanie wokół osi poziomej przechodzącej przez środek odcinka daje obraz klepsydry. Usunięcie niewidocznych linii daje figurę trójwymiarową o realistycznym kształcie.

10.10. Napisz program wyświetlania figury obrotowej trójwymiarowej utworzonej z paraboli. Wyświetlona parabola może najpierw być obracana wokół osi pionowej (równoległej do osi Y), przechodzącej przez jej środek, o małe kroki kątowe od położenia początkowego aż do 180°. Następnie obracanie wokół osi poziomej (równoległej do osi X) przechodzącej przez parabolę umożliwi zajrzenie do jej wklęsłego wnętrza. Łącząc liniami końce obróconych paraboli otrzymasz kształt pucharu.

10.11. Narysuj trójwymiarowy wykres słupkowy na środku ekranu, a następnie przesuń go, obróć i skorzystaj z rzutów perspektywicznych, aby pokazać wykres w dowolnym położeniu na ekranie.

10.12. Opracuj ogólny program trójwymiarowych transformacji zawierający różne transformacje, rzuty perspektywiczne i metodę eliminacji niewidocznych linii (lub powierzchni) jako oddzielne podprogramy. Zaplanuj możliwość wybrania przez użytkownika dowolnych transformacji dla wprowadzonego przez niego obiektu i wyświetlania obiektu w perspektywie z usuniętymi niewidocznymi liniami. Dopuść możliwość kolejnych transformacji aż do chwili, gdy użytkownik zechce zakończyć program (pisząc na przykład STOP).

10.13. Zmień program z poprzedniego zadania tak, aby kod literowy napisany u dołu (u góry lub z boku) ekranu określał wybór opcji, łącznie z zakończeniem, gdy zostanie wprowadzony z klawiatury. Jako alternatywa wyboru z klawiatury może być użyty wybór za pomocą pióra świetlnego.

Część V
ZASTOSOWANIA
(Jak możemy korzystać z grafiki?)

W części I przedstawiliśmy przegląd zastosowań graficznych. Teraz, po przedstawieniu różnych metod realizacji obrazów graficznych, przyjrzymy się bardziej szczegółowo zastosowaniom. Najpierw rozważymy metody skutecznego projektowania programów graficznych, a następnie przedyskutujemy zastosowania grafiki w przedsiębiorstwie, nauczaniu i w domu.

11. Projekt programu

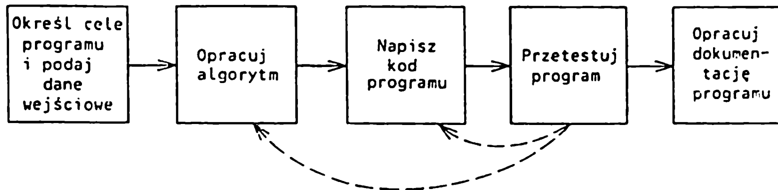
W trakcie opracowywania programów do zastosowań graficznych często stwierdzamy, że przekroczyliśmy pojemność pamięci komputera, że program wykonuje się zbyt długo lub, że jest zbyt trudny do modyfikacji czy uruchamiania. Jeśli nie poświęcimy dostatecznie wiele uwagi na projekt programu, to możemy oczekiwać, że napotkamy wymienione problemy tym częściej, im nasz program będzie bardziej złożony. Dzięki zastosowaniu standardowych metod projektowania programów przed rozpoczęciem ich pisania możemy uniknąć wielu problemów, a programy staną się łatwiejsze do adaptacji dla różnorodnych zastosowań. W tym rozdziale przedstawimy niektóre metody poprawnego projektowania programów i ich zastosowanie do programów graficznych.

11.1. Organizacja programów graficznych

Zaczniemy od przedstawienia podstawowych kroków opracowania programów. Następnie rozważymy, jak efektywnie planować i organizować programy w każdym z tych kroków. Ponadto, efektywność programu wyrażona wielkością zajmowanego obszaru pamięci lub czasem wykonania może być ważnym czynnikiem wpływającym na jego projekt.

Fazy opracowania programu

Przy opracowaniu programu można wyróżnić 5 faz (rys. 11.1). Musimy: 1) zdefiniować i określić główne części programu; 2) zdecydować się na ciąg instrukcji (algorytm), jaki zastosujemy w programie; 3) napisać ten ciąg instrukcji (zakodować program); 4) przetestować program i 5) przekazać w programie pewne wyjaśnienia (dokumentację) dotyczące jego działania. Możemy oszczędzić czas i uniknąć kłopotów, jeśli każdemu z tych kroków poświęcimy trochę uwagi w trakcie opracowywania programu.



Rys. 11.1. Fazy opracowywania programu

Definiowanie głównych części programu oznacza, że musimy dokładnie określić, co program ma robić i jak zamierzamy dostarczyć dane dla niego. Jeśli mamy zamiar utworzyć obraz, to musimy zdefiniować, co ma zawierać każda jego część. Jeśli tworzymy wykres, to decydujemy się na jego typ, orientację i rozmieszczenie na ekranie. Dane do tworzenia obrazu mogą być zdefiniowane w instrukcjach typu DATA, wprowadzone przez instrukcję INPUT, obliczone z równań lub uzyskane z kombinacji tych metod. Dane wprowadza się do programu na początku lub przewiduje się w programie konwersacyjne ich wprowadzanie. Określenie wszystkich zmiennych wejściowych, dokładnej postaci użytych danych i głównych metod przetwarzania (obliczanie, sortowanie, skalowanie danych do wymiarów ekranu itp.) uprości programiście opracowanie szczegółowego algorytmu i rozkazów w programie. W tym miejscu musimy pamiętać o późniejszych zmianach i modyfikacjach, które zechcemy zrobić na obrazie i powinniśmy to przewidzieć przy definiowaniu programu.

Po zarysowaniu celu programu i zdefiniowaniu niezbędnych danych, możemy określić bezpośrednio kroki przetwarzania w każdej części programu. Jest to faza opracowywania algorytmu. Tutaj też możemy ustalić ogólną strukturę naszego programu zgodnie z metodami *programowania strukturalnego*. Metody te umożliwiają taką organizację programu, aby testowanie, uruchamianie i — w razie potrzeby — modyfikowanie programu było łatwe.

W zasadzie, metody programowania strukturalnego wymagają *modułowego projektu* programu bez zbędnych rozgałęzień utrudniających śledzenie jego przebiegu. W oddzielnych *modułach* ustalamy typ niezbędnego przetwarzania (wejście, skalowanie danych, transformacje, rysowanie). Moduły te mogą być podprogramami lub po prostu oddzielnymi blokami kodu. Celem tworzenia modułów jest założenie aby w jednej części programu nie łączyć różnych prac: każdy moduł wykonuje tylko jedną czynność. Mamy więc jeden moduł (lub część programu) do wprowadzania danych, inny moduł do sortowania danych itd. Później można łatwo określić, co każda część programu wykonuje — jeśli poszukujemy błędu lub chcemy dokonać zmian. Podobnie, za pomocą oddzielnych modułów można zdefiniować i wyświetlić każdy składnik wykresu (osie, słupki, krzywe). Umożliwia to, łatwe usunięcie lub dodanie części rysunku. W końcowym etapie ustawiamy moduły w takim porządku, w jakim mają być przetwarzane w programie. Moduły, które mają

być używane przez inne moduły – takie jak na przykład podprogram kreślenia – umieszczamy na końcu programu. Dopuszcza się wejście do modułu tylko na jego początku i wyjście z niego tylko na końcu. Unika się dzięki temu wielokrotnych instrukcji GOTO lub RETURN, które mogą skomplikować program. Lista kolejności modułów lub jakaś postać sieci działań umożliwi pokazanie poszczególnych kroków przetwarzania.

Kodowanie w Basicu (lub innym języku programowania) może być teraz wykonane zgodnie z planem naszego algorytmu – moduł po module. W celu umożliwienia późniejszych manipulacji w obrazie należy tak kodować, aby wszystkie wyświetlane punkty były zapamiętane w tablicach. Później łatwo jest dodać moduł transformacji przetwarzający zadaną tablicę punktów. Nazwy zmiennych najlepiej jest tak wybrać, aby były mnemonikami parametrów, które reprezentują, i należy konsekwentnie używać tych samych nazw w kolejnych programach. Należy stosować nazwy: dla zmiennych czasowych T, dla kąta A, a dla współrzędnych X i Y. Upraszcza to śledzenie kroków programu, zwłaszcza po napisaniu kilku różnych programów. Odpowiednio dobrany system rozmieszczania tekstu programu bardzo ułatwia czytanie go. Na przykład możemy „wciąć" wszystkie instrukcje między instrukcjami FOR i NEXT w celu identyfikacji głównej części pętli. Podobnie możemy „wciąć" inne części kodu, na przykład wewnętrzne instrukcje IF – także w celu ułatwienia czytania programu. Niestety, w niektórych systemach następuje automatyczne usuwanie początkowych spacji w wierszach instrukcji, co uniemożliwia ich wcinanie.

W celu uproszczenia procesu uruchamiania testuje się program moduł po module. Jeśli nawet nie jest to możliwe, to nadal chcemy przekonać się, czy każdy moduł poprawnie wykonuje swą pracę. Jeśli nasz program miał tylko wyświetlić określony rysunek lub wykres, to łatwo na tej podstawie zdecydować, czy pracuje poprawnie. Program jest poprawny, jeśli otrzymany rysunek odpowiada naszym żądaniom. Łatwo także można określić, co trzeba zmienić, jeśli rysunek nie jest poprawny. Programy bardziej uniwersalne, na przykład do wyświetlania dowolnej krzywej, powinny mieć przetestowane wszystkie możliwe drogi logiczne. Wymaga to testowania dla różnych krzywych w celu przekonania się, czy są one rysowane we właściwym miejscu ekranu, czy osie i ich oznaczenia są prawidłowo umieszczone. Jeżeli jakaś część programu nie pracuje tak jak trzeba, to najpierw sprawdzamy, czy kod odpowiada algorytmowi. Często wystarczy poprawić kod programu, ale może się też okazać, że trzeba wrócić do fazy opracowywania programu w celu poprawienia algorytmu. Testowanie powinno odbyć się z dostateczną liczbą danych wejściowych, aby sprawdzić poprawność działania każdego modułu.

Dokumentowanie programu jest procesem ciągłym we wszystkich fazach opracowywania programu. Określenie celu programu, algorytmu i konwencji kodowania wchodzą w całości w skład dokumentacji. Możemy przewidzieć „plik dokumentacyjny", zawierający definicje programu i algorytmu w celu późniejszego odwołania się do niego. Oprócz tego, użycie opisowych nazw zmiennych, odpowiedni układ typograficzny, modułowy projekt programu oraz liczne komentarze w tekście

programu (instrukcje REM) mogą uczynić nasz program praktycznie samodokumentującym się.

Efektywność programu

Ograniczenia dotyczące pojemności pamięci i czasu wykonywania programu są następnymi czynnikami wpływającymi na organizację programu. Mogą one być w konflikcie z celami programu ustalonymi zgodnie z metodami programowania strukturalnego.

Jeśli czynnikiem ograniczającym jest pojemność pamięci, to możemy być zmuszeni do zrezygnowania z czytelności programu i jego samodokumentacji dla oszczędności pamięci. Możemy naturalnie zawsze podzielić program tak, że przetwarzanie graficzne jest wykonywane częściami przez więcej niż jeden program. Możemy także skrócić program eliminując powtarzające się instrukcje lub części programu. Może to oznaczać potrzebę łączenia modułów lub segmentów programu, niszczącą w pewnym sensie modułowość programu. Program można skrócić także usuwając zbędne spacje w liniach programu i zbędne komentarze (instrukcje REM), umieszczając wiele instrukcji w jednej linii. Wszystkie te metody prowadzą do oszczędności pojemności pamięci, ale pogarszają czytelność programu, utrudniają jego uruchamianie i modyfikację.

Ograniczenia czasowe mają mniejszy wpływ na strukturę programu i na ogół nie są brane pod uwagę, chyba że zachodzi potrzeba animacji obrazu. W celu przyśpieszenia wykonania programu wystarczy zmniejszyć liczbę obliczeń. Dla animacji oznacza to przesuwanie mniejszej liczby części na większe odległości.

Istnieje wiele sposobów zmniejszania liczby obliczeń. Każde obliczenia zbędne eliminuje się. Jeśli wykonaliśmy jakieś obliczenie raz, to upewniamy się, czy nie jest powtarzane — jak niekiedy może zdarzyć się w pętlach

```
610  FOR I = 1 TO N
620       FOR J = 1 TO 1000
630            K = I + 3
```

W tym przypadku obliczamy $K = I + 3$ tysiąc razy dla każdej spośród N wartości I. Aby uniknąć tego typu redundancji, instrukcja podstawienia w linii 630 powinna być przeniesiona przed linię 620. Funkcje trygonometryczne i wykładnicze wymagają długiego czasu obliczeń; często możemy tak zmienić program, aby nie korzystać z tych funkcji. W ogólności, opracowanie algorytmu wymagającego jedynie dodawania i odejmowania liczb całkowitych znacznie przyspieszy obliczenia.

Liczbę obliczeń możemy także zmniejszyć częściej korzystając z instrukcji warunkowych IF w celu ominięcia zbędnych obliczeń. Zwykle zwiększa to jednak

rozmiar programu. Eliminacja lub zmniejszenie wymiarów tablic także oszczędza czas. Ponadto więcej czasu trzeba do obliczenia adresu elementu tablicy dwuwymiarowej niż jednowymiarowej. Podobnie dostęp do elementów tablicy zajmuje więcej czasu niż dostęp do zmiennej prostej. Inna metoda zmniejszenia czasu wykonywania programów polega na napisaniu ich w asemblerze lub języku kompilowanym — nie w interpretowanym.

Niektóre systemy mikrokomputerowe są wyposażone w specjalne funkcje graficzne, które stosuje się do zwiększania szybkości wykonywania programów. Zwykle są to funkcje animacji obrazów, operacje transformacji i okna.

11.2. Metody konwersacyjne

Efektywna metoda wprowadzania danych do programów graficznych polega na dialogu. Za pomocą tej metody program może prowadzić ograniczony dialog z operatorem pytając go, co chciałby następnie zrobić. Wybór przedstawia się w postaci zestawu opcji (menu), a wyboru jednej z opcji można dokonać za pomocą jednego z wielu urządzeń wejściowych. Do naszej dyspozycji stoi wiele interakcyjnych urządzeń wejściowych. Najpopularniejsze to: klawiatura, pióro świetlne, tabliczka graficzna i manipulatory.

Wybór z menu

Lista opcji przetwarzania prezentowana przez program jest zwana *menu*. Wiele przedstawionych w tej książce programów używało menu jako sposobu wyboru: położenia na ekranie, metody transformacji lub elementów obrazu, które mają podlegać transformacji. Za każdym razem, gdy projektujemy ogólny program graficzny z opcjami przetwarzania, możemy korzystać z menu do wyboru opcji.

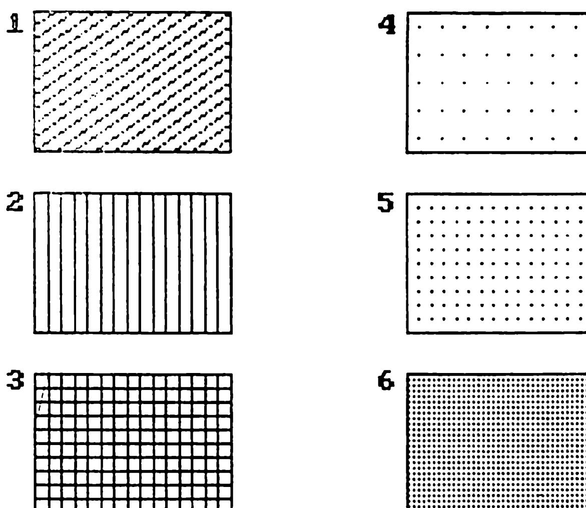
```
Wybierz obraz do wyświetlenia:
        1 — wóz strażacki
        2 — chłopiec z psem
        3 — klown
        4 — samolot
        5 — robot
        6 — żaglówka
        7 — ryba
```

Rys. 11.2. Wybór opcji z menu z ponumerowanymi opcjami

Rodzaj menu zależy od typu urządzenia wejściowego stosowanego do współpracy z programem.

W dotychczasowych programach zakładaliśmy, że współpraca z programem odbywała się jedynie przez klawiaturę. W takim przypadku numeryczna lub alfanumeryczna lista opcji (rys. 11.2) jest prostym i skutecznym sposobem wyboru opcji. Napisanie litery lub cyfry określającej opcję powoduje przejście programu do modułu wykonującego tę opcję (np. modułu rysującego klowna, jeśli wybierzemy opcję nr 3). Tę metodę wyboru możemy stosować do przedstawiania wielu różnych typów opcji programowych. Na rysunku 11.3 pokazano wybór z menu różnych wzorów cieniowania, a na rys. 11.4 – listę możliwych obliczeń do wykonania na

Wybierz wzór:

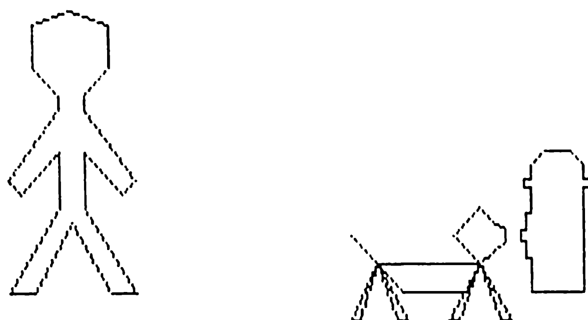


Rys. 11.3. Menu do wyboru wzoru cieniowania

```
Jakie obliczenia wykonać?

      S: Obliczenie średniej i wariancji
      L: Aproks. metodą najmn. kwadratów
      Q: Aproksymacja kwadratowa
      C: Aproksymacja trzeciego rzędu
      G: Test błędów aproksymacji
```

Rys. 11.4. Menu opcji przetwarzania danych wejściowych



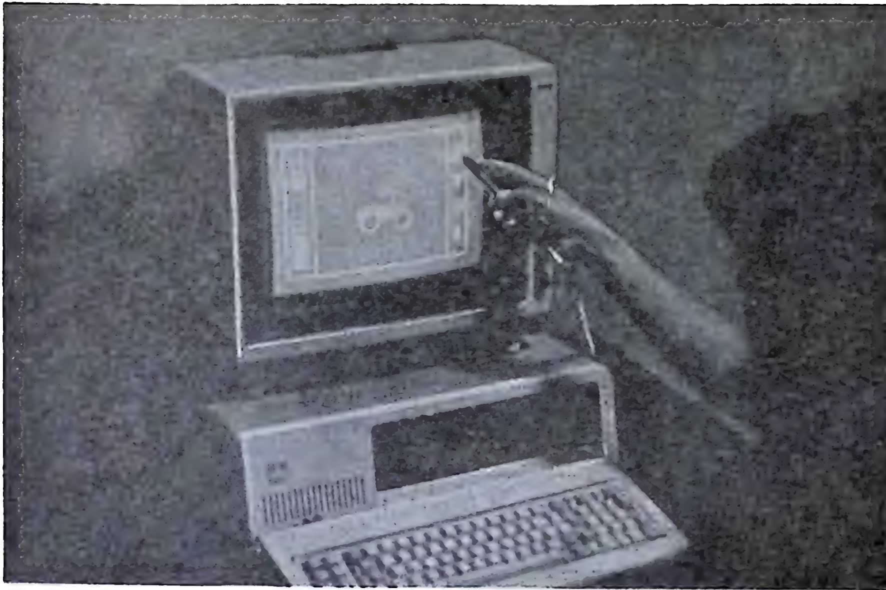
Rys. 11.5. Jednoczesne wyświetlanie menu i rysunku

danych wejściowych. Do wyboru opcji z listy można użyć dowolnego klawisza z klawiatury, ale na ogół do większości zastosowań wskazane jest ułożenie opcji w porządku alfabetycznym lub numerycznym.

Menu może zapełniać cały ekran lub może być umieszczone z boku (na górze, na dole). Jeśli menu zapełnia cały ekran, to będziemy musieli je usunąć po dokonaniu wyboru. Umieszczenie menu na niewielkim obszarze ekranu umożliwia połączenie rysunku i menu (rys. 11.5). W tym przypadku menu pozostaje na ekranie nawet po dokonaniu wyboru opcji. Jest to szczególnie użyteczne wówczas, gdy opcje są ciągle dostępne, gdyż nie wymaga przerysowywaniu rysunku po każdym wyborze opcji. Po dokonaniu wszystkich selekcji można usunąć menu. Jednoczesna prezentacja rysunku i menu jest użyteczna, gdy korzystamy z różnych typów urządzeń do współpracy z programem oraz gdy chcemy wyświetlić kilka menu. Wadą jest zmniejszenie użytecznej powierzchni ekranu o część zajętą przez menu.

Pióro świetlne

Na rysunku 11.6 pokazano użycie pióra świetlnego do wyboru opcji z menu. Pióro świetlne ma kształt ołówka i wykrywa światło pochodzące z określonego punktu na ekranie monitora. Światło to (poświata luminoforu pokrywającego ekran) pojawia się wskutek przejścia strumienia elektronów. Jeśli uaktywnione pióro świetlne wskazuje punkt na ekranie w chwili, gdy punkt ten jest rozjaśniony przez padający strumień elektronów, to możliwe jest zapamiętanie ekranowych współ-



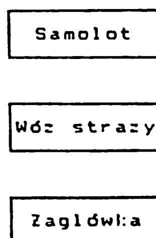
Rys. 11.6. Wybór z menu za pomocą pióra świetlnego

rzędnych tego punktu. Ponieważ strumień elektronów przebiega przez każdy punkt ekranu co najmniej 25 razy na sekundę, wykrycie rozjaśnionego punktu jest prawie natychmiastowe.

Pióro świetlne uaktywnia się wieloma sposobami. Niektóre pióra mają włącznik przyciskowy uaktywniony przez dociśnięcie końcówki pióra świetlnego do ekranu. Inne mają przycisk z boku pióra wciskany w celu uaktywnienia pióra. I wreszcie, w innym typie pióra w pobliżu jego końca znajduje się metalowy pierścień, którego dotykanie uaktywnia pióro. Wiele systemów mikrokomputerowych ma w języku Basic instrukcje do testowania, czy pióro świetlne zostało uaktywnione, i do odczytu jego położenia na ekranie. Gdy pióro jest uaktywnione, aktualne położenie pióra może być użyte do wyboru opcji z menu, rysowania punktu czy linii lub przesuwania wyświetlonych obiektów.

Wybór z menu za pomocą pióra świetlnego realizuje się w programie przez sprawdzanie wartości współrzędnych otrzymanych z pióra do określenia wybranego obiektu. Wybiera się ten obiekt, w którego prostokątnym otoczeniu mieszczą się współrzędne pióra. Przypuśćmy, że współrzędne pióra wskazują punkt leżący w prostokącie z napisem WÓZ STRAŻY (rys. 11.7). Do stwierdzenia, że został wybrany prostokąt z napisem WÓZ STRAŻY, wystarczy sprawdzenie współrzędnej Y. Opcje wskazywane przez menu przedstawione na rys. 11.7 powinny być umieszczane daleko od siebie, aby ich wybór był jednoznaczny. Wyboru można dokonać używa-

Wskaż piórem obiekt do narysowania:



Rys. 11.7. Uaktywnienie pióra świetlnego ustawionego wewnątrz prostokąta z napisem spowoduje zapamiętanie jego współrzędnych

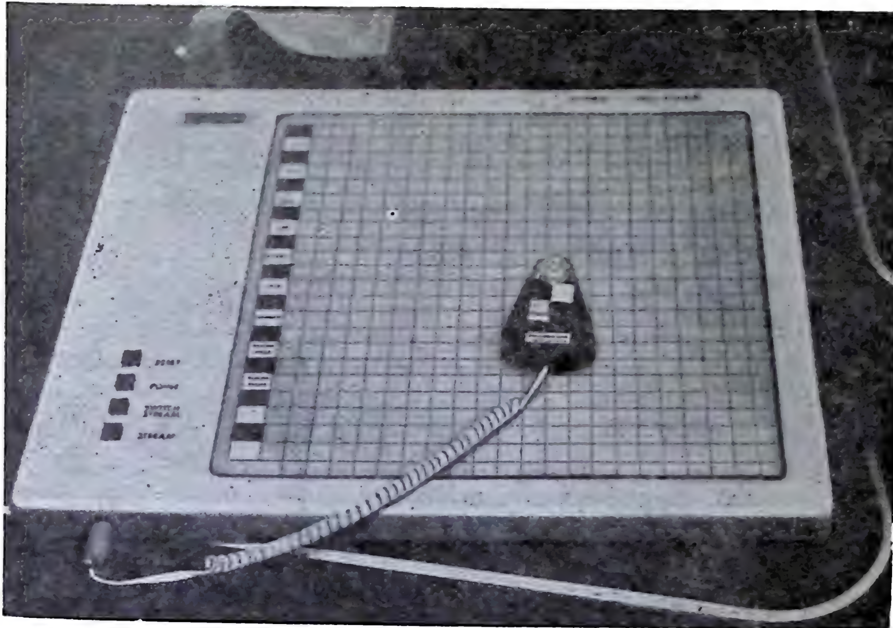
jąc słów, kolorowych prostokątów (przy określeniu kolorów) lub kształtów (przy wyborze obiektów).

Pióro świetlne stosuje się także do innych celów niż tylko wybór z menu. Za pomocą programu w Basicu można piórem świetlnym rysować linię ciągłą na podstawie kolejnych współrzędnych lub linię złożoną z odcinków linii prostych. Umożliwia to rysowanie obiektów i rysunków na ekranie, ale wymaga precyzyjnego prowadzenia pióra. Z metody tej korzystamy do przeniesienia na ekran rysunku z przezroczystej folii lub kalki technicznej. Programowo da się taki rysunek zapamiętać w pliku, który może być następnie przetwarzany lub łączony z innymi plikami, na przykład z tekstem. Możemy także napisać program rysowania obiektu o zadanym kształcie (linia, prostokąt, wielokąt, okrąg) w miejscu wyznaczonym przez pióro świetlne. Stosowanie pióra świetlnego ułatwia manipulację obiektami na ekranie – przesuwanie, skalowanie lub obrót. Obiekt przesuwamy uaktywniając pióro w pozycji, do której obiekt ma być przesunięty. Skalowanie i obrót wymagają wyboru opcji z menu.

Pióro świetlne musi wykrywać światło z ekranu do zapisania współrzędnych i wobec tego – w przypadku większości komputerów osobistych – nie może być stosowane do wskazywania położenia na czarnym tle, trzeba więc wybrać inne kolor tła. W niektórych systemach przez wszystkie pozycje na ekranie stale przybiega mały świecący kursor i wtedy pióro może być używane nawet na czarnym tle.

Tablica graficzna

Tablica graficzna (ang. *graphic tablet*), podobnie jak pióro świetlne, może służyć do wyboru położenia na ekranie. Teraz jednak wybieramy pozycję na powierzchni tablicy, a nie na ekranie. Początek układu współrzędnych (0, 0) na tablicy jest wybrany w tym samym punkcie co na ekranie, tj. w lewym, górnym rogu. W pamięci zostaną zapamiętane współrzędne punktu tablicy, nad którym umieszczono uaktywniony ręczny kursor lub wskaźnik. Wskaźnik uaktywnia się przez naciśnięcie przycisku. Na kursorze znajduje się zwykle kilka przycisków, co umożliwia wpro-



Rys. 11.8. Tablica graficzna z kursorem

wadzanie pojedynczych punktów lub ciągu punktów przy ruchu kursora po tablicy. Na zdjęciu 11.8 pokazano tablicę z kursorem.

Położenie punktu na tablicy wybieramy ustawiając skrzyżowane linie „celownika" kursora nad tym punktem. W wielu tablicach współrzędne określa się na podstawie pomiaru napięcia na siatce przewodów na powierzchni tablicy. Napięcie w każdym przewodzie jest inne, tak że różnica napięć w jakimś kierunku odpowiada różnicy współrzędnych. Aktywacja kursora lub wskaźnika powoduje zapisanie wartości napięcia w tym miejscu. Wartość tę przekształca się następnie we współrzędne X, Y i zapamiętuje. W niektórych tablicach do określenia położenia używa się fali ultradźwiękowej; takie tablice zawierają wzdłuż dwóch brzegów mikrofony liniowe (paskowe). Aktywacja wskaźnika powoduje generację iskry dającej dźwięk, który rozchodzi się do brzegów tablicy. Czas przejścia dźwięku do jednego i do drugiego brzegu tablicy odpowiada współrzędnym X i Y. Przy odczytywaniu współrzędnych z tablicy korzysta się zwykle z programu w języku wewnętrznym. Taki program można wywołać z Basicu przez instrukcję CALL. Programy te będą różne dla różnych typów tablic.

Interakcyjne posługiwanie się tablicą graficzną jest podobne do sposobu korzystania z pióra świetlnego. Tablica umożliwia jednak odczytywanie współrzędnych znacznie dokładniej niż pióro świetlne, lepiej więc nadaje się do wprowadzania

na ekran wykresów, map lub rysunków. Po wprowadzeniu dane te można zapamiętać do późniejszego użycia. Ustawianie położenia obiektów przesuwanych wykonuje się przez wprowadzenie nowego położenia obiektu. Za pomocą tablicy możemy rysować na ekranie rysunki lub wybierać opcje z menu. Do wyboru opcji z menu zwykle jest potrzebna przezroczysta folia z nadrukiem menu. Kształty obiektów z menu nanosi się na tę folię i umieszcza ją potem na powierzchni tablicy. Wybór punktów wewnątrz obszaru narysowanego obiektu wskazuje ten obiekt. Podobnie można zaprojektować nadruki na folli do wyboru koloru.

Manipulatory

Do większości mikrokomputerów można przyłączyć manipulator. Manipulator ma zwykle kształt dźwigni (ang. *joystick*), którą można przesuwać w lewo, w prawo, w górę i w dół. W większości systemów w Basicu przewidziano instrukcje przekształcenia ruchów manipulatora we współrzędne. Jeżeli manipulator wskazuje pewien kierunek, to są generowane współrzędne dla tego kierunku. Za pomocą manipulatora możemy przesuwać w sposób ciągły obiekty na ekranie. Ruchy manipulatora można także przekształcić we współczynniki skali lub kąt obrotu obiektów. Podobnie jak w przypadku pióra świetlnego lub tablicy graficznej, niektóre manipulatory są wyposażone w jeden lub więcej przycisków lub przełączników. Umożliwia to posługiwanie się manipulatorem przy wyborze z menu lub rysowaniu kształtów.

11.3. Konstrukcja obrazu

Techniki konwersacyjne umożliwiają tworzenie obrazu fragmentami. W elementarnej postaci takie techniki stosowaliśmy w programie 3.18, w którym tworzyliśmy obraz linia po linii. Ta metoda krokowego tworzenia obrazu może być rozszerzona przez włączenie innych elementarnych kształtów i ich transformacji. Można więc utworzyć pewną liczbę „cegiełek" i składać z nich obraz. Interakcyjnie można taki kształt przenieść w pożądane miejsce dokonując na nim wybranych transformacji. W programie 11.1 przedstawiono zarys metody tworzenia obrazu za pomocą linii, trójkątów, prostokątów i okręgów. Program prezentuje dostępne kształty jako menu z lewej strony ekranu, na dole zaś pokazuje dostępne transformacje. Migotanie słowa WYBIERZ określa, że należy dokonać wyboru z menu. Na rysunku 11.9 przedstawiono widok ekranu.

Program 11.1 składa się z modułów wykonujących poszczególne zadania. Wyboru z menu dokonuje się za pomocą klawiatury. Przez dodanie rozkazów sprawdzenia aktywacji pióra świetlnego możemy ten program zaadaptować do pracy z piórem. Sprawdzenie wartości współrzędnych wskazywanych przez pióro umożliwi określenie, który element został wybrany. Trzecie menu umożliwi wybór wielkości

Program 11.1. Interakcyjne tworzenie obrazu przy użyciu podstawowych kształtów (linia, trójkąt, prostokąt, okrąg) wybieranych z menu

```
10  'PROGRAM 11.1. Składanie rysunku z części
20      'Program wyświetla na ekranie wykaz dopuszczalnych opcji
30      'kształtów i transformacji. Do każdego wybranego kształtu
40      'można zastosować dowolną liczbę transformacji spośród
50      'dopuszczalnych transformacji.
60      '*******************************************************
70  CLEARSCREEN
80  YA = 5/12                      'YA i XA określają rozdzielczość
90  XA = 12/5
100 XM = 639
110 YM = 199
120 GRAPHICS
130 GOSUB 830                      'Narysuj oba menu
140 AA = 0                         'AA jest całkowitym kątem obrotu
150     '**************** Wprowadź wybór ***********************
160 POSITION 1,1
170 PRINT "        ";              'Migocący napis WYBIERZ
180 FOR K = 1 TO 300:NEXT K
190 POSITION 1,1
200 PRINT "WYBIERZ:";
210 FOR K = 1 TO 300:NEXT K
220 A$ = INKEY$                    'A$ określa wybrany kształt
230 IF A$ = "" OR A$ < "1" OR A$ > "4" THEN 160
240 RO = VAL(A$) * 4
250 N = 0                          'N wskazuje, czy wybrany kształt
260 POSITION 23,1                  'powinien być usunięty, czy nie
270 PRINT "WYBIERZ:";              'Migocący napis WYBIERZ i nr kształtu
280 POSITION RO,1
290 PRINT " ";
300 FOR K = 1 TO 300:NEXT K
310 POSITION 23,1
320 PRINT "        ";
330 POSITION RO,1
340 PRINT A$;
350 FOR K = 1 TO 300:NEXT K
360 B$ = INKEY$                    'B$ określa wybraną  transformację
370 IF B$ = "" OR B$<>"P" AND B$<>"S" AND B$<>"O" AND B$<>"W" AND
                                   B$<>"N" AND B$<>"K" THEN 260
380 IF B$ = "K" THEN 3110
390 IF B$ <> "N" THEN 440
400 RESTORE                        'Wróć do początku danych
410 GOSUB 990
420     '********** Usunąć? *******
430 GOTO 140
440 IF B$ <> "W" THEN 530
450 COLOR 0,0                   'Niech kolor rysunku bqdzie taki jak kolor tła
460 IF B$ = "W" AND A$ = "1" THEN GOSUB 1070
470 IF B$ = "W" AND A$ = "2" THEN GOSUB 1100
480 IF B$ = "W" AND A$ = "3" THEN GOSUB 1160
490 IF B$ = "W" AND A$ = "4" THEN GOSUB 1430
500 RESTORE
510 GOSUB 990
520 GOTO 140
530 POSITION 23,1: PRINT "         ";
540 POSITION 24,1: PRINT "
                            ";
550     '************* Przesunąć? ***********
560 IF B$ <> "P" THEN 640
570 GOSUB 1850                     'Wypisz instrukcje do przesunięcia
580 COLOR 0,0     'Aby wymazać, wybierz kolor rysunku taki jak kolor tła
590 IF B$ = "P" AND A$ = "1" THEN GOSUB 1510
600 IF B$ = "P" AND A$ = "2" THEN GOSUB 1590
610 IF B$ = "P" AND A$ = "3" THEN GOSUB 1690
```

Program 11.1 (cd.)

```
620 IF B$ = "P" AND A$ = "4" THEN GOSUB 1760
630     '************* Skalować? ***********
640 IF B$ <> "S" THEN 720
650 GOSUB 2330                   'Wypisz instrukcje do skalowania
660 COLOR 0,0
670 IF B$ = "S" AND A$ = "1" THEN GOSUB 1920
680 IF B$ = "S" AND A$ = "2" THEN GOSUB 2020
680 IF B$ = "S" AND A$ = "3" THEN GOSUB 2140
700 IF B$ = "S" AND A$ = "4" THEN GOSUB 2220
710     '************* Obrócić? ***********
720 IF B$ <> "O" THEN 790
730 GOSUB 3000                   'Wypisz instrukcje do obrotu
740 COLOR 0,0
750 IF B$ = "O" AND A$ = "1" THEN GOSUB 2400
760 IF B$ = "O" AND A$ = "2" THEN GOSUB 2540
770 IF B$ = "O" AND A$ = "3" THEN GOSUB 2740
780 IF B$ = "O" AND A$ = "4" THEN GOSUB 2830
790 N = 1                        'Od tej chwili usuń stary kształt
800 GOSUB 960                    'Wyświetl znowu menu transformacji
810 GOTO 260
820     '
830     '************** Wyświetl oba menu ************************
840 POSITION 1,1: PRINT "WYBIERZ:";
850 POSITION 4,1: PRINT "1.";
860 POSITION 8,1: PRINT "2.";
870 POSITION 12,1: PRINT "3.";
880 POSITION 16,1: PRINT "4.";
890 COLOR 1,0
900 GOSUB 990
910 GOSUB 1080
920 GOSUB 1100
930 AA = 0
940 GOSUB 1160
950 GOSUB 1430
960 POSITION 23,1: PRINT "WYBIERZ:
 ";
970 POSITION 24,1: PRINT "P -Przesuń   S -Skaluj  O -Obróć   W -Wymaż
 N -Następny obiekt   K -Koniec";
980 RETURN
990     '******************* Odczytaj dane ***************************
1000 READ XP,YP,XQ,YQ
1010 READ XU,YU,XV,YV,XW,YW,XX,YX
1020 READ XC,YC,RX,RY
1030 READ XR,YR,XS,YS,XT,YT
1040 RETURN
1050     '
1060     '********************* Program rysowania *******************
1070      '******* Rysuj linię **********
1080 DRAWLINE XP,YP TO XQ,YQ
1090 RETURN
1100      '******* Rysuj prostokąt ******
1110 DRAWLINE XU,YU TO XV,YV
1120 DRAWLINE XV,YV TO XW,YW
1130 DRAWLINE XW,YW TO XX,YX
1140 DRAWLINE XX,YX TO XU,YU
1150 RETURN
1160      '******* Rysuj okrąg **********
1170 IF RX < RY THEN 1200
1180 R = RY
1190 GOTO 1210
1200 R = RX
1210 IF AA <> 0 THEN 1310    'Figura była obracana. Wybierz coś innego
1220 FOR A = 0 TO 1.5708 STEP 1/R       'Kreśl 4 symetryczne punkty
```

Program 11.1 (cd.)

```
1230   DX = RX * COS(A)
1240   DY = RY * SIN(A)
1250   POINTPLOT XC+DX,YC+DY*YA
1260   POINTPLOT XC+DX,YC-DY*YA
1270   POINTPLOT XC-DX,YC+DY*YA
1280   POINTPLOT XC-DX,YC-DY*YA
1290 NEXT A
1300 GOTO 1420
1310 CX = SIN(AA) * XA         'Oblicz stałą część zależności
1320 CY = SIN(AA) * YA
1330 FOR A = 0 3.14159 STEP 1/R
1340   X = RX * COS(A)          'Równanie okręgu
1350   Y = RY * SIN(A) * YA
1360   XH = X
1370   X = X * COS(AA) + Y * CX
1380   Y = Y * COS(AA) - XH * CY
1390   POINTPLOT XC+X,YC+Y
1400   POINTPLOT XC-X,YC-Y
1410 NEXT A
1420 RETURN
1430      '******** Rysuj trójkąt *******
1440 DRAWLINE XR,YR TO XS,YS
1450 DRAWLINE XS,YS TO XT,YT
1460 DRAWLINE XT,YT TO XR,YR
1470 RETURN
1480    '
1490    '****************** Przesunięcie ************************
1500      '******** Przesuń linię ********
1510 IF N = 0 THEN 1530                'Jeśli kształt jest ciągle w menu,
1520 GOSUB 1070                        'to nie usuwaj go
1530 XP = XP + HT: YP = YP + VT
1540 XQ = XQ + HT: YQ = YQ + VT
1550 COLOR 1,0
1560 GOSUB 1070                        'Wymaż
1570 RETURN
1580      '******* Przesuń prostokąt *****
1590 IF N = 0 THEN 1610
1600 GOSUB 1100                        'Rysuj
1610 XU = XU + HT: YU = YU + VT
1620 XV = XV + HT: YV = YV + VT
1630 XW = XW + HT: YW = YW + VT
1640 XX = XX + HT: YX = YX + VT
1650 COLOR 1,0
1660 GOSUB 1100                        'Rysuj
1670 RETURN
1680      '******* Przesuń okrąg **********
1690 IF N = 0 THEN 1710
1700 GOSUB 1160                        'Wymaż
1710 XC = XC + HT: YC = YC + VT
1720 COLOR 1,0
1730 GOSUB 1160                        'Rysuj
1740 RETURN
1750      '******* Przesuń trójkąt ********
1760 IF N = 0 THEN 1780
1770 GOSUB 1430                        'Wymaż
1780 XR = XR + HT: YR = YR + VT
1790 XS = XS + HT: YS = YS + VT
1800 XT = XT + HT: YT = YT + VT
1810 COLOR 1,0
1820 GOSUB 1430                        'Rysuj
1830 RETURN
1840      '*********** Instrukcje ********
1850 POSITION 23,1: PRINT "        ";
```

Program 11.1 (cd.)

```
1860 POSITION 23,1
1870 PRINT " Określ wielkość przesunięcia w poziomie i w pionie?";
1880 INPUT HT, VT
1890 RETURN
1900   '
1910   '*********************** Skalowanie ****************************
1920     '******* Skalowanie linii ******
1930 IF N = 0 THEN 1950
1940 GOSUB 1070                        'Wymaż
1950 XF = (XP + XQ) / 2
1960 YF = (YP + YQ) / 2
1970 XP = XP * HS + XF (1 - HS): YP = YP * VS + YF (1 - VS):
1980 XQ = XQ * HS + XF (1 - HS): YQ = YQ * VS + YF (1 - HS):
1990 COLOR 1,0
2000 GOSUB 1070                        'Rysuj
2010 RETURN
2020     '******* Skalowanie prostokąta ******
2030 IF N = 0 THEN 2050
2040 GOSUB 1100                        'Wymaż
2050 XF = (XU + XV + XW + XX) / 4
2060 YF = (YU + YV + YW + YX) / 4
2070 XU = XU * HS + XF (1 - HS): YU = YU * VS + YF (1 - VS):
2080 XV = XV * HS + XF (1 - HS): YV = YV * VS + YF (1 - VS):
2090 XW = XW * HS + XF (1 - HS): YW = YW * VS + YF (1 - VS):
2100 XX = XX * HS + XF (1 - HS): YX = YX * VS + YF (1 - VS):
2110 COLOR 1,0
2120 GOSUB 1100                        'Rysuj
2130 RETURN
2140     '******* Skalowanie okręgu *********
2150 IF N = 0 THEN 2170
2160 GOSUB 1160                        'Wymaż
2170 RX = RX * HS
2180 RY = RY * VS
2190 COLOR 1,0
2200 GOSUB 1160                        'Rysuj
2210 RETURN
2220     '******* Skalowanie trójkąta *********
2230 IF N = 0 THEN 2250
2240 GOSUB 1430                        'Wymaż
2250 XF = (XR + XS + XT) / 3
2260 YF = (YR + YS + YT) / 3
2270 XR = XR * HS + XF (1 - HS): YR = YR * VS + YF (1 - VS):
2280 XS = XS * HS + XF (1 - HS): YS = YS * VS + YF (1 - VS):
2290 XT = XT * HS + XF (1 - HS): YT = YT * VS + YF (1 - VS):
2300 COLOR 1,0
2310 GOSUB 1430                        'Rysuj
2320 RETURN
2330   '*************** Instrukcje **********************
2340 POSITION 23,1
2350 PRINT "Wprowadź współczynniki skali X i Y";
2360 INPUT HS, VS
2370 RETURN
2380   '
2390   '*********************** Obrót **********************************
2400     '************ Obrót linii *********
2410 IF N = 0 THEN 2430
2420 GOSUB 1070                        'Wymaż
2430 XO = (XO + XP) / 2                'Znajdź środek linii
2440 YO = (YO + YP) / 2
2450 XH = XP             'Zapamiętaj wartość XP do dalszych obliczeń
2460 XP = XO + (XP - XO) * COS(AR) + (YP - YO) * SIN(AR) * XA
2470 YP = YO + (YP - YO) * COS(AR) - (XH - XO) * SIN(AR) * YA
2480 XH = XQ
```

Program 11.1 (cd.)

```
2490 XQ = XO + (XQ - XO) * COS(AR) + (YQ - YO) * SIN(AR) * XA
2500 YQ = YO + (YQ - YO) * COS(AR) - (XH - XO) * SIN(AR) * YA
2510 COLOR 1,0
2520 GOSUB 1430                    'Rysuj
2530 RETURN
2540     '************ Obrót prostokąta *********
2550 IF N = 0 THEN 2570
2560 GOSUB 1100                    'Wymaż
2570 XO = (XU + XV + XW + XX) / 4  'Znajdź środek prostokąta
2580 YO = (YU + YV + YW + YX) / 4
2590 XH = XU          'Zapamiętaj wartość XU do dalszych obliczeń
2600 XU = XO + (XU - XO) * COS(AR) + (YU - YO) * SIN(AR) * XA
2610 YU = YO + (YU - YO) * COS(AR) - (XH - XO) * SIN(AR) * YA
2620 XH = XV
2630 XV = XO + (XV - XO) * COS(AR) + (YV - YO) * SIN(AR) * XA
2640 YV = YO + (YV - YO) * COS(AR) - (XH - XO) * SIN(AR) * YA
2650 XH = XW
2660 XW = XO + (XW - XO) * COS(AR) + (YW - YO) * SIN(AR) * XA
2670 YW = YO + (YW - YO) * COS(AR) - (XH - XO) * SIN(AR) * YA
2680 XH = XX
2690 XX = XO + (XX - XO) * COS(AR) + (YX - YO) * SIN(AR) * XA
2700 YX = YO + (YX - YO) * COS(AR) - (XH - XO) * SIN(AR) * YA
2710 COLOR 1,0
2720 GOSUB 1100                    'Rysuj
2730 RETURN
2740     '************ Obrót okręgu  ************
2750 IF N = 0 THEN 2780
2760 AA = AS
2770 GOSUB 1160                    'Wymaż
2780 AA = AR + AS
2790 COLOR 1,0
2800 GOSUB 1160                    'Rysuj
2810 AS = AA
2820 RETURN
2830     '************ Obrót trójkąta ************
2840 IF N = 0 THEN 2860
2850 GOSUB 1430                    'Wymaż
2860 XO = (XR + XS + XT) / 3       'Znajdź środek trójkąta
2870 YO = (YR + YS + YT) / 3
2880 XH = XR
2890 XR = XO + (XR - XO) * COS(AR) + (YR - YO) * SIN(AR) * XA
2900 YR = YO + (YR - YO) * COS(AR) - (XH - XO) * SIN(AR) * YA
2910 XH = XS
2920 XS = XO + (XS - XO) * COS(AR) + (YS - YO) * SIN(AR) * XA
2930 YS = YO + (YS - YO) * COS(AR) - (XH - XO) * SIN(AR) * YA
2940 XH = XT
2950 XT = XO + (XT - XO) * COS(AR) + (YT - YO) * SIN(AR) * XA
2960 YT = YO + (YT - YO) * COS(AR) - (XH - XO) * SIN(AR) * YA
2970 COLOR 1,0
2980 GOSUB 1430                    'Rysuj
2990 RETURN
3000   '*************** Instrukcje *************************
3010 POSITION 23,1
3020 PRINT "O ile stopni obrócić?";
3030 INPUT AR
3040 AR = AR * 3.14159 / 180       'Zamień na radiany
3050 RETURN
3060   '***************************************************
3070 DATA 30,27,80,27
3080 DATA 80,45,80,70,30,70,30,45
3090 DATA 55,90,22,22
3100 DATA 30,130,80,130,55,110
3110 END
```

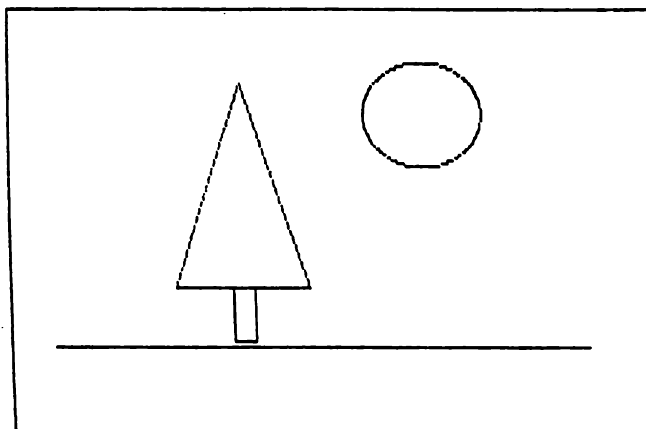
WYBIERZ:

1. ——

2. □

3. ○

4. △



P–Przesuń S–Skaluj O–Obróć W–Wymaż
N–Następny obiekt K–Koniec

Rys. 11.9. Wyświetlanie dwóch menu i rysunek utworzony przez program 11.1

przesunięcia, obrotu i współczynników skali. Może ono być zrealizowane w postaci wyskalowanego odcinka (odcinków). Do wyboru kąta obrotu odcinek powinien być wyskalowany od 0 do 360°. Do przesunięcia linia będzie wyskalowana od wartości ujemnych do dodatnich w dopuszczalnym zakresie.

Tworzenie obrazu krokami, tak jak w programie 11.1, jest w wielu zastosowaniach dobrą metodą ogólną. Można korzystać z niej w aplikacyjnych programach, przy rysowaniu wykresów, wyborze wzorów graficznych, grach lub przy eksperymentach z różnymi kształtami.

12. Zastosowania grafiki w gospodarce

Jeden z największych i najbardziej różnorodnych zakresów zastosowań grafiki to zastosowania w gospodarce. Generowane za pomocą komputera wykresy, rysunki i obrazy powszechnie stosuje się do wspomagania analiz finansowych, studiów marketingowych oraz w planowaniu i podejmowaniu decyzji. Są to typowe zastosowania grafiki w wielu różnych przedsiębiorstwach. Wykresy i rysunki dostarczają informacji o budżecie, stanie zapasów materiałowych, przepływach kapitałów, dochodzie, oprocentowaniu pożyczek, amortyzacji inwestycji i portfelach akcji. Za pomocą wykresów porównuje się ceny lub charakterystyki produktów różnych producentów, pokazuje lokalne tendencje handlowe lub zależności ilości sprzedanych towarów dla różnych regionów kraju, w rozliczeniu na poszczególnych sprzedawców lub kolejne lata. Wykresy danych demograficznych pomagają zlokalizować potencjalnych klientów lub dostosować podział terytoriów sprzedaży. Diagramy czasowe lub wykresy działań wspomagają synchronizację projektów. Obrazowe przedstawienie rozmieszczenia przestrzennego ułatwia planowanie usytuowania sprzętu i działów pomocniczych w przedsiębiorstwie.

Wykresy i rysunki mogą przedstawiać dane w podziale geograficznym lub dla poszczególnych wydziałów przedsiębiorstwa. W celu pokazania przyszłych tendencji lub szybkiego porównania możemy graficznie skorelować dane obecne z poprzednimi. Z takich wykresów korzysta się w raportach wewnętrznych i prezentacjach dla klientów. W tym rozdziale przedstawimy niektóre metody tworzenia wykresów i rysunków dla celów gospodarczych.

12.1. Metody ogólne

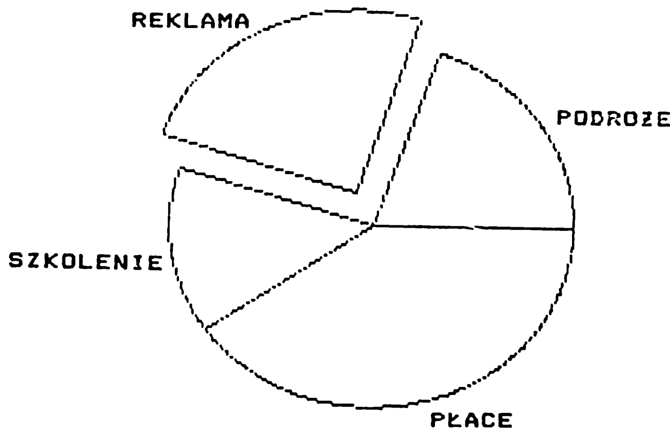
Metody rysowania wykresów omówiono w rozdz. 4 i 5. Przedstawimy teraz pewne rozszerzenia tych metod graficznych do zastosowań gospodarczych.

Program 12.1. Wykres kołowy

```
10  'PROGRAM 12.1. Rozszerzony wykres kołowy
20      'Program rysuje opisany wykres kołowy z wysuniętym wycinkiem.
30  CLEARSCREEN
40  DIM N$(8), V(8)
50  PRINT "Podaj maksymalne wartości współrzędnych X i Y"
60  PRINT "ekranu dla wybranej rozdzielczości"
70  INPUT XM, YM
80  PRINT "Podaj liczbę znaków jaka może być napisana"
90  PRINT "na ekranie przy tej rozdzielczości"
100     '
110 INPUT C1
120 PC = XM / C1             'PC jest liczbą pikseli na znak w poziomie
130 PRINT "Podaj liczbę wierszy tekstu, jaka mieści się"
140 PRINT "na ekranie przy tej rozdzielczości"
150 INPUT R1
160 PR = YM / R1             'PR jest liczbą pikseli na znak w pionie
170 PRINT
180 PRINT "Podaj współrzędne środka wykresu kołowego"
190 INPUT XN, YN
200 PRINT "Podaj promień"
210 INPUT R
220 IF XN+R > XM OR XN-R < 0 OR YN+R > YM OR YN-R < 0 THEN 1020
230 PRINT
240 .    '****************** Wprowadź dane ***********************
250 PRINT "Podaj liczbę wycinków (nie więcej niż 8)"
260 INPUT N
270 PRINT "Podaj nazwę i wartość każdego wycinka"
280 T = 0
290     'Wprowadź dane. Znajdź sumę (T) wszystkich wartości"
300 FOR K = 1 TO N
310   INPUT N$(K), V(K)
320   T = T + V(K)
330 NEXT K
340 PRINT "Który wycinek wysunąć? (1 -"; N; ")";
350 INPUT E
360 IF E < 1 OR E > N THEN 340
370     '*************** Narysuj wykres kołowy ****************
380 CLEARSCREEN
390 GRAPHICS
400 YA = 5 / 6       'YA dopasowuje wykres do wybranej rozdzielczości
410 B = 0            'B jest kątem nachylenia poprzedniej linii
420 S = 0
430 RE = 360 * 3.14159 /180     'RE jest równe 2π
440 FOR K = 1 TO N
450   XC = XN                   'XN, YN - współrzędne normalnego środka
460   YC = YN
470   'Rysowana linia jest określona przez wartość aktualnego
480   'wycinka oraz sumę (S) poprzednich
490   S = S +V(K)
500   'Kąt linii wycinka jest równy S/T razy 360 stopni
510   '
520   A = RE * S / T
530   IF K <> E THEN 660        'Czy ten wycinek ma być wysunięty?
540   AC = B + (A - B) / 2      'AC jest kątem dwusiecznej wycinka
550   XE = XC + R/5 * COS(AC)   'Tak. Nowy środek w p. (XE,YE), w odl.
560   YE = YC + R/5 * SIN(AC) * YA    '1/5 dł. promienia od środka wykresu
570   XP = XE + R * COS(B)            'Znajdź końce linii i narysuj wycinek
580   YP = YE + R * SIN(B) * YA
590   DRAWLINE XE,YE TO XP,YP
600   XP = XC + R * COS(A)
610   YP = YC + R * SIN(A) * YA
620   DRAWLINE XC,YC TO XP,YP
630   XC = XE
```

Program 12.1 (cd.)

```
640   YC = YE
650   '
660   FOR A1 = B TO A STEP 1/R
670     XP = XC + R * COS(A1)
680     YP = YC + R * SIN(A1) * YA
690     POINTPLOT XP,YP
700   NEXT A1
710   DRAWLINE XC,YC TO XP,YP
720   'Opisz wycinek
730      'Znajdź punkt odległy o 4 znaki od środka łuku
740      'wycinka, na zewnątrz, po promieniu
750   AC = B + (A - B) / 2     'AC jest kątem dwusiecznej wycinka
760   XL = XC + (R + 4) * COS(AC)
770   YL = YC + (R + 4) * SIN(AC) * YA
780      'Punkt (XL,YL) oznacza miejsce zaczepienia opisu wycinka
790      'Użyj tego punktu jako początku napisu - jeśli jest po prawej
800      'stronie wycinka, jako końca - jeśli jest po lewej i jako
810      'środka - jeśli jest na górze lub na dole
820   'Punkt (XL,YL) jest początkiem napisu
830   IF XL > XC + 10 THEN 930
840   'Punkt (XL,YL) jest końcem napisu
850   IF XL < XC - 10 THEN 900
860   'Punkt (XL,YL) jest środkiem napisu. Przesuń XL o połowę liczby
870   'pikseli potrzebnych na napis
880   XL = XL - LEN(N$(K)) / 2 * PC
890   GOTO 930
900   'Punkt (XL,YL) jest końcem napisu. Przesuń do tyłu o liczbę
910   'pikseli potrzebnych na napis
920   XL = XL - LEN(N$(K)) * PC
930   'Sprowadź (XL,YL) do najbliższej pozycji znakowej, na której
940   'można umieścić napis
950   RO = INT(YL / PR) + 1
960   CO = INT(XL / PC) + 1
970   POSITION RO,CO
980   PRINT N$(K);
990   B = A
1000 NEXT K
1010 GOTO 1030
1020 PRINT "Współrzędne poza dopuszczalnym zakresem"
1030 END
```



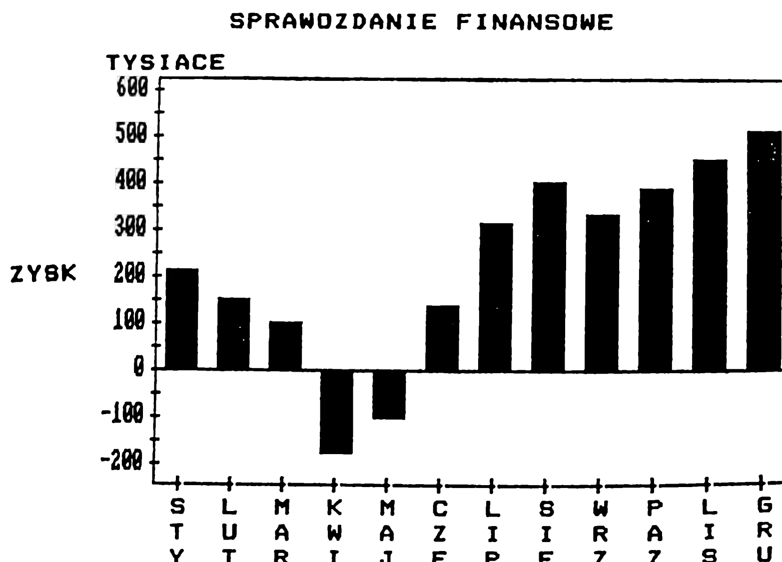
Rys. 12.1. Rozszerzony wykres kołowy – narysowany przez program 12.1

Jedna z metod prezentacji polega na przedstawieniu wykresu kołowego z wyróżnionym jednym lub dwoma wycinkami przez wysunięcie ich po promieniu. Przykład takiego rozszerzonego wykresu kołowego przedstawiono na rys. 12.1. Technika tworzenia takiego wykresu stosowana w programie 12.1 jest kontynuacją metody dyskutowanej w p. 5.4, dotyczącej sposobu sytuowania opisów wykresu kołowego. Najpierw obliczamy kątowe granice wycinka, który zostanie wysunięty, a następnie kąt nachylenia promienia przechodzącego przez środek tego wycinka. W końcu, znajdujemy środek okręgu wysuniętego wycinka znajdujący się na prostej łączącej środek wycinka i środek wykresu kołowego i rysujemy ten wycinek. W programie 12.1 wycinki wysuwa się na odległość równą 1/5 długości promienia wykresu kołowego.

Istnieje niekiedy potrzeba narysowania wykresu pokazującego wielkości ujemne, na przykład straty finansowe. Możemy je przedstawić rysując wielkości ujemne poniżej osi poziomej (rys. 12.2). Wykresy te tworzy się rozciągając oś pionową poniżej osi poziomej i opisując wartości ujemne i dodatnie.

Wykreślenie dwóch typów wykresów na tej samej osi jest skuteczną metodą skondensowania wykresu lub porównania danych. Program 12.2 zawiera przykład kreślenia wykresu słupkowego na jednej części osi poziomej i wykresu liniowego na drugiej części. Na osi poziomej użyto różnego skalowania obu części wykresu; oś pionowa ma wspólne skalowanie dla obu części wykresu. Ten sposób umożliwia przedstawianie dwóch zbiorów danych w różnych postaciach (rys. 12.3).

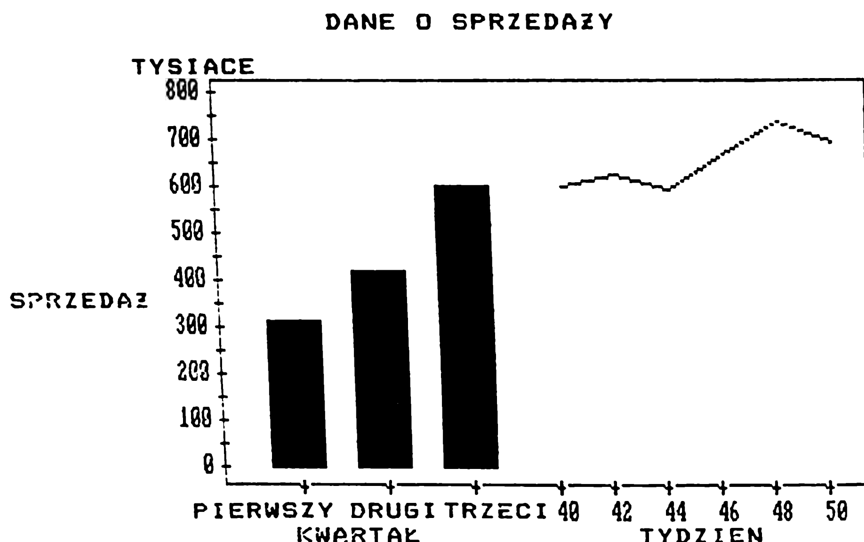
Przedstawianie danych w różnych postaciach graficznych może ułatwić interpretację danych. Program 12.3 jest przykładem projektu modułowego, umożli-



Rys. 12.2. Wykres słupkowy zawierający dane ujemne i dodatnie

Program 12.2. Połączenie różnych postaci wykresu: wykres słupkowy i liniowy

```
10  'PROGRAM 12.2. Wykres liniowy i słupkowy
20       'Wykres liniowy i słupkowy opisane na osi X ze wspólnym
30       'opisem na osi Y. Zakładamy, że ekran ma wymiary 640 x 200
30       'pikseli, co daje 25 wierszy po 80 znaków. Dane są skalowane
40       'tak, aby mieściły się między liniami 28 a 156.
50       '
60       '***********************************************************
70  DIM X(12),Y(12)
80  CLEARSCREEN
90  GRAPHICS
100 PRINT
110 PRINT TAB(34); "DANE O SPRZEDAZY"
120 DRAWLINE 128,24 TO 128,162
130 DRAWLINE 128,162 TO 608,162
140 DRAWLINE 608,162 TO 608,24
150 DRAWLINE 608,24 TO 128,24
160      'Narysuj podziałki na osi Y
170 FOR Y = 28 TO 156 STEP 8
180   DRAWLINE 125,Y TO 131,Y
190 NEXT Y
200      ,Oznakuj oś Y
210 RO = 20
220 FOR S = 0 TO 800 STEP 100
230   POSITION RO,13
240   PRINT USING "###"; S
250   RO = RO - 2
260 NEXT S
270      'Narysuj podziałki na osi X
280 FOR CO = 24 TO 40 STEP 8
290   POSITION 21,CO
300   PRINT "+";
310 NEXT CO
320 FOR CO = 48 TO 73 STEP 5
330   POSITION 21,CO
340   PRINT "+"
350 NEXT CO
360 PRINT TAB(22);"PIERWSZY DRUGI    TRZECI    40    42    44    48    50"
370 PRINT TAB(22);"     KWARTAŁ                         TYDZIEN"
380   '
390   '
400 POSITION 13,3
410 PRINT "SPRZEDAZ"
420      'Narysuj słupki wykresu
430 T = (156 - 28) / 800
440 X1 = 163
450 FOR K = 1 TO 3
460   READ S
470   Y = INT((800 - S) * T + 28.5)
480   FOR X = X1 TO X1 + 40
490     DRAWLINE X,Y TO X,156
500   NEXT X
510   X1 = X1 + 64
520 NEXT K
530 X = 384
540 FOR K = 1 TO 6
550   READ S
560   Y(K) = INT((800 - S) * T + 28.5)
570 NEXT K
580 FOR K = 1 TO 5
590   DRAWLINE X,Y(K) TO X+40,Y(K+1)
600   X = X + 40
610 NEXT K
620 DATA 310,420,599,598,623,592,670,740,695
630 END
```



Rys. 12.3. Połączenie różnych postaci wykresu o wspólnej skali pionowej i różnych skalach poziomych uzyskane za pomocą programu 12.2.

Program 12.3. Ogólny program rysowania wykresów — umożliwia wybór typu wykresu

```
10  'PROGRAM 12.3. Ogólny program rysowania diagramów
20       'Program umożliwia konwersacyjne wprowadzanie danych
30       'i wybór typu diagramu (liniowy, słupkowy, kołowy), jest
40       'przystosowany do ekranu 640 x 200 pikseli.
50       '***********************************************************
60  PC = 8                 'PC jest liczbą pikseli znaku w poziomie
70  PR = 8                 'PR jest liczbą pikseli znaku w pionie
80  YA = 5/12    'YA dopasowuje wykres do wybranej rozdzielczości
90  CLEARSCREEN
100 PRINT "1 - LINIOWY    2 - SŁUPKOWY     3 - KOŁOWY"
110 PRINT "Jaki rodzaj wykresu?"
120 INPUT C$
130 PRINT "Jaki tytuł wykresu?"
140 INPUT T$
150 GRAPHICS
160 IF C$ = "1" OR C$ = "2" THEN GOSUB 210
170 IF C$ = "1" THEN GOSUB 630
180 IF C$ = "2" THEN GOSUB 740
190 IF C$ = "3" THEN GOSUB 860
200 GOTO 1520
210 '*************** Wykres liniowy lub słupkowy ******************
220 PRINT "Podaj liczbę części"
230 INPUT N
240 D = INT(568 / N)
250 PRINT "Podaj nazwę i wartość każdej części wykresu"
260 FOR K = 1 TO N
270   INPUT L$(K), S(K)
280   IF LEN(L$(K)) < D/8 THEN 320       'Czy napis się zmieści?
290   PRINT "Nazwa za długa. Maksymalna długość wynosi ";D/8
300   PRINT "Nowa nazwa?";
```

Program 12.3 (cd.)

```
310   INPUT L$(K)
320 NEXT K
330 PRINT "Podaj wartości minimalną i maksymalną dla osi Y"
340 INPUT LO, HI
350 R = HI - LO                   'R jest zakresem wartości Y
360 RS = (180 - 20) / R           'RS służy do skalowania
370 CLEARSCREEN
380 DRAWLINE 71,20 TO 71,180
390 DRAWLINE 71,180 TO 639,180
400 DRAWLINE 639,180 TO 639,20
410 DRAWLINE 639,20 TO 71,20
420 POSITION 1,1                  'Napisz tytuł wykresu
430 P = 40 - LEN(T$) / 2          'Ustaw tytuł w środku
440 PRINT TAB(P);T$
450 RO = 23
460 Y = 180
470 FOR K = 0 TO 5                'Podziel zakresu wartości wykresu
480   POSITION RO,1               'w pionie na 5 części
490   L = LO + R * K / 5
500   PRINT USING "###.###";L
510   DRAWLINE 70,Y TO 639,Y
520   RO = RO - 4
530   Y = Y - 32
540 NEXT K
550      'Opisz oś poziomą
560 POSITION 25,1
570 FOR K = 1 TO N
580   P = INT((72 +(K-1) * D + D/2) / PC - LEN(L$(K)/2 + 0.5) + 1
590   PRINT TAB(F);L$(K);
600 NEXT K
610 RETURN
620 '
630 '****************** Wykres liniowy ************************
640 FOR K = 1 TO N
650   Y(K) = INT((HI - S(K) * RS + 20.5)
660 NEXT K
670 X = 71 + D/2
680 FOR K = 1 TO N-1
690   DRAWLINE X,Y(K) TO X+D,Y(K+1)
700   X = X + D
710 NEXT K
720 RETURN
730 '
740 '****************** Wykres słupkowy *************************
750 YO = INT(HI * RS + 20.5)            'Znajdź środek
760 X1 = 71 + D/6
770 FOR K = 1 TO N
780   Y = INT((HI - S(K) * RS + 20.5)
790   FOR X = X1 TO X1 + D * 2/3
800     DRAWLINE X,Y TO X,YO
810   NEXT X
820   X1 = X1 + D
830 NEXT K
840 RETURN
850 '
860 '****************** Wykres kołowy *************************
870 XC = 320
880 YC = 110
890 R = 160
900 PRINT "Jaka liczba wycinków (nie więcej niż 6)?"
910 INPUT N
920 PRINT "Podaj nazwę i wartość każdego wycinka"
930 T = 0
940      'Wprowadź dane. Znajdź sumę (T) wszystkich wycinków
```

Program 12.3 (cd.)

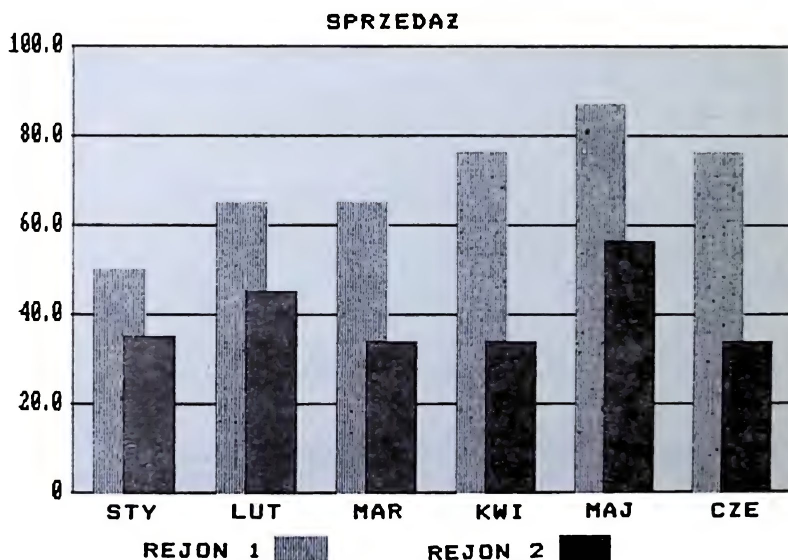
```
950 FOR K = 1 TO N
960    INPUT N$(K), V(K)
970    IF V(K) > 0 THEN 1010
980    PRINT "Wartość musi być dodatnia. Wprowadź nową wartość"
990    INPUT V(K)
1000  GOTO 970
1010  T = T + V(K)
1020 NEXT K
1030 CLEARSCREEN
1040 GRAPHICS
1050 POSITION 1,1
1060 P = 40 r LEN(T$) / 2              'Ustaw tytuł w środku
1070 PRINT TAB(P);T$
1080 CIRCLEPLOT XC,YC,R
1090 B = 0                'B jest kątem nachylenia poprzedniej linii
1100 S = 0
1110 RE = 360 * 3.14159 / 180          'RE wynosi 2π
1120 FOR K = 1 TO N
1130        'Rysowana linia jest określona przez wartość aktualnego
1140        'wycinka i sumę (S) porzednich
1150    S = S + V(K)
1160        'Kąt linii wycinka jest równy S/T razy 360 stopni
1170        '
1180    A = RE * S / T
1190    XP = XC + R * COS(A)
1200    YP = YC + R * SIN(A) * YA
1210    DRAWLINE XC,YC TO XP,YP
1220    'Opisz wycinki
1230        'Znajdź punkt odległy o 4 jednostki od środka łuku
1240        'wycinka, na zewnątrz, po promieniu
1250    AC = B + (B-A)/2  'AC jest nachyleniem dwusiecznej wycinka
1260    XL = XC + (R + 4) * COS(AC)
1270    YL = YC + (R + 4) * SIN(AC) * YA
1280      'Punkt (XL,YL) oznacza miejsce zaczepienia opisu wycinka
1290      'Użyj tego punktu jako początku napisu - jeśli jest po prawej
1300      'stronie wycinka, jako końca - jeśli jest po lewej i jako
1310      'środka - jeśli jest na górze lub na dole
1320    'Punkt (XL,YL) jest początkiem napisu
1330    IF XL > XC + 10 THEN 1430
1340    'Punkt (XL,YL) jest końcem napisu
1350    IF XL < XC - 10 THEN 1400
1360    'Punkt (XL,YL) jest środkiem napisu. Przesuń XL o połowę liczby
1370    'pikseli potrzebnych na napis
1380    XL = XL - LEN(N$(K)) / 2 * PC
1390    GOTO 1430
1400    'Punkt (XL,YL) jest końcem napisu. Przesuń do tyłu o liczbę
1410    'pikseli potrzebnych na napis
1420    XL = XL - LEN(N$(K)) * PC
1430    'Sprowadź (XL,YL) do najbliższej pozycji znakowej, na której
1440    'można umieścić napis
1450    RO = INT(YL / PR) + 1
1460    CO = INT(XL / PC) + 1
1470    POSITION RO,CO
1480    PRINT N$(K);
1490    B = A
1500 NEXT K
1510 RETURN
1520 END
```

wiającego wybór postaci wykresu oraz przedstawianych danych. Ten ogólny projekt może być użyteczny przy tworzeniu różnych wykresów tych samych danych i umożliwia eksperymentowanie z różnymi parametrami.

12.2. Wykresy porównawcze

Na jednym wykresie możemy wykreślić dwa lub więcej zbiorów danych dla porównania zależności między zmiennymi, na przykład sprzedaży w różnych regionach lub w zależności od sprzedawcy. Takie wykresy porównawcze tworzy się różnymi sposobami.

Na wykresie słupkowym możemy porównać dwa zbiory danych przez częściowe nałożenie słupków (rys. 12.4). Program 12.4 tworzy taki porównawczy wykres słupkowy kreśląc słupki dla różnych zbiorów danych w różnych kolorach. Słupki jednego zbioru danych są lekko przesunięte w stosunku do słupków drugiego zbioru i pokrywają je częściowo.



Rys. 12.4. Porównawczy wykres słupkowy utworzony przez program 12.4, pokazujący nakładające się słupki dla dwóch zbiorów danych, wykreślone w tych samych osiach współrzędnych

Program 12.4. Wykresy porównawcze: nałożone wykresy słupkowe

```
10  'PROGRAM 12.4. Dwa nałożone wykresy słupkowe
20      'Program rysuje dwa zbiory danych w postaci wykresów słupkowych
30      'ze wspólnym opisem osi. Zakładamy wymiary ekranu 640 x 200
40      'pikseli. Dane są skalowane od linii 12 do 172.
50      '************************************************************
60  CLEARSCREEN
70  DIM L$(8), S1(8), S2(8)
80  PC = 8                     'PC jest liczbą pikseli na znak w poziomie
90      '********** Wprowadź tytuł i dane do rysunku ****************
```

Program 12.4 (cd.)

```
100 PRINT "Podaj tytuł rysunku"
110 INPUT T$
120 PRINT "Podaj nazwę pierwszego zbioru danych";
130 INPUT T1$
140 PRINT "Podaj nazwę drugiego zbioru danych";
150 INPUT T2$
160 PRINT "Podaj liczbę części"
170 INPUT N
180 D = INT(568 / N)                    'D jest obszarem każdej części
190 PRINT 'Podaj nazwę i wprowadź dwie wartości do każdej części"
200 FOR K = 1 TO N
210   INPUT L$(K), S1(K), S2(K)
220   IF LEN(L$(K)) < D/PC THEN 260     'Czy nazwa się zmieści?
230   PRINT "Nazwa za długa. Maksymalna długość wynosi ";D/PC
240   PRINT "Podaj nową nazwę";
250   INPUT L$(K)
260 NEXT K
270 PRINT "Podaj wartość minimalną i maksymalną dla osi Y"
280 INPUT LO, HI
290 R = HI - LO                         'R jest zakresem wartości Y
300 RS = (172 - 12) / R                 'RS służy do skalowania
310      '*********** Rysuj siatkę i opisy ************
320 CLEARSCREEN
330 GRAPHICS
340 DRAWLINE 72,12 TO 71,172
350 DRAWLINE 71,172 TO 639,172
360 DRAWLINE 639,172 TO 639,12
370 DRAWLINE 639,12 TO 72,12
380 POSITION 1,1                        'Napisz tytuł rysunku
390 P = 40 - LEN(T$) / 2                'Ustaw tytuł w środku
400 PRINT TAB(P);T$
410 RO = 22
420 Y = 172
430 FOR K = 1 TO 5                      'Podziel zakresu wartości
440   POSITION RO,3                     'rysunku w pionie na 5 części
450   L = LO + R * K / 5
460   PRINT USING "####.#";L
470   DRAWLINE 70,Y TO 639,Y
480   RO = RO - 4
490   Y = Y - 32
500 NEXT K
510      'Opisz części
520 POSITION 23,1
530 FOR K = 1 TO N
540   P = INT((71+(K-1)*D+D/2)/PC - LEN(L$(K)/2 + 0.5) + 1
550   PRINT TAB(P);L$(K);
560 NEXT K
570      'Oznacz kodami kolorowymi zbiory danych
580 POSITION 25,28-LEN(T1$)
590 PRINT T1$;
600 COLOR 7,0                           'Narysuj pomarańczowy prostokąt
610 FOR X = 228 TO 268
620   DRAWLINE X,190 TO X,199
630 NEXT X
640 POSITION 25,28-LEN(T2$)
650 PRINT T2$;
660 COLOR 2,0                           'Narysuj niebieski prostokąt
670 FOR X = 452 TO 492
680   DRAWLINE X,190 TO X,199
690 NEXT X
700      '*************** Narysuj słupki ****************
710 YO = INT(HI * RS +12.5)             'Znajdź środek
```

Program 12.4 (cd.)

```
720 X1 = 71 + D/6
730 FOR K = 1 TO N
740   Y = INT((HI - S1(K)) * RS + 12.5)
750   COLOR 7,0                             'Rysuj pomarańczowe słupki
760   FOR X = X1 TO TO X1 + D * 5/12
770     DRAWLINE X,Y TO X,YO
780   NEXT X
790   X2 = X1 + D * 3/12
800   Y = INT((HI - S2(K)) * RS + 12.5)
810   COLOR 2,0                             'Rysuj niebieskie słupki
820   FOR X = X2 TO TO X2 + D * 5/12
830     DRAWLINE X,Y TO X,YO
840   NEXT X
850   X1 = X1 + D
860 NEXT K
870 END
```

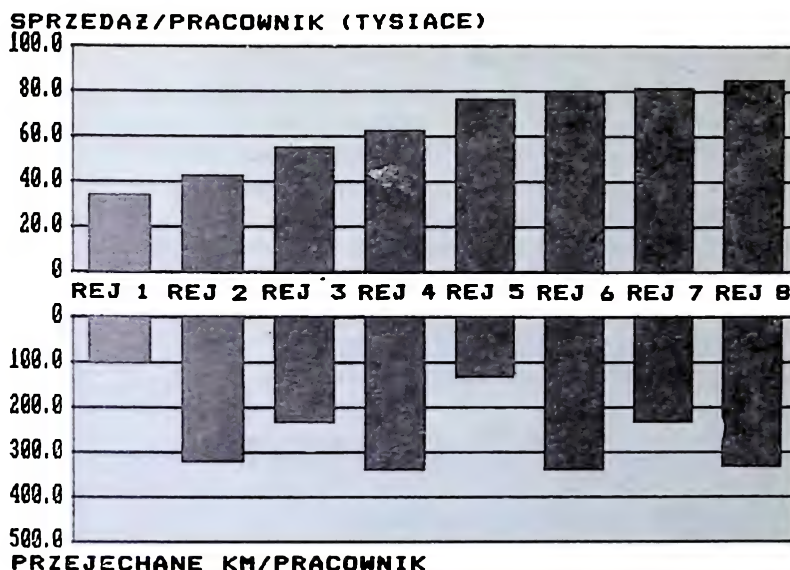
Inny przykład wykresu porównawczego przedstawiono na rys. 12.5. Wykres ten został narysowany przez program 12.5. W tym przykładzie dwa zbiory danych mają tę samą oś X, ale różne osie Y. Umożliwia to wprowadzenie różnego skalowania osi pionowej dla różnych zbiorów danych.

Program 12.5. Porównawcze wykresy słupkowe: jedne słupki w dół, drugie w górę

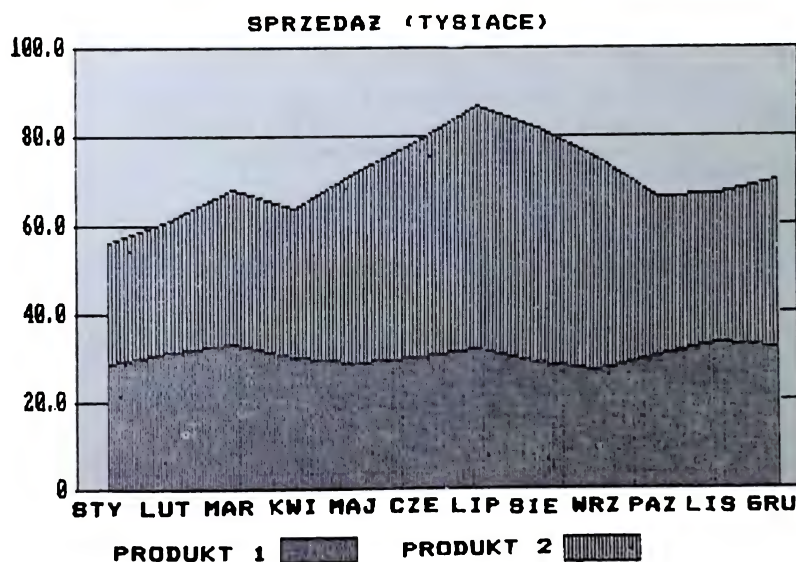
```
10  'PROGRAM 12.5. Wykres porównawczy z dwoma skalami Y
20      'Program umożliwia wprowadzenie tytułów i dwóch zbiorów danych
30      'o dowolnej liczbie części. Słupki jednego zbioru danych są
40      'rysowane w górnej części ekranu, a słupki drugiego zbioru
50      'rysowane są w dolnej części ekranu. Program jest przysto-
60      'sowany do ekranu o wymiarach 640 x 200 pikseli.
70      '
80      '*************************************************************
90  CLEARSCREEN
100 DIM L4(15), T(15), B(15)
110 PC = 8                    'PC jest liczbą pikseli na znak w poziomie
120     '********************** Wprowadź dane ***********************
130 PRINT "Podaj liczbę części"
140 INPUT N
150 D = INT(568 / N)           'D jest liczbą pikseli na część wykresu
160 PRINT "Podaj nazwę i wartość każdej części"
170 FOR K = 1 TO N
180   INPUT L$(K), T(K), B(K)
190   IF LEN(L$(K)) < D/PC THEN 230     'Czy nazwa się zmieści?
200   PRINT "Nazwa za długa. Maksymalna długość wynosi ";D/PC
210   PRINT "Podaj nową nazwę";
220   INPUT L$(K)
230 NEXT K
240 PRINT "Podaj opis górnej części osi Y"
250 INPUT T$
260 PRINT "Podaj wartości minimalną i maksymalną dla górnej części wykresu"
270 INPUT LT, HT
280 R1 = HT - LT
290 RT = (92 - 12) / R1     'RT służy do skalowania górnej części wykresu
300 PRINT "Podaj opis dołu osi Y"
310 INPUT B$
320 PRINT "Podaj wartości minimalną i maksymalną dla dolnej części wykresu"
330 INPUT LB, HB
340 R2 = HB - LB
```

Program 12.5 (cd.)

```
350 RB = (188 - 108) / R2    'RB służy do skalowania dolnej części wykresu
360      '**************** Narysuj tło **********************
370 CLEARSCREEN
380 GRAPHICS
390 DRAWLINE 71,12 TO 71,92
400 DRAWLINE 71,92 TO 639,92
410 DRAWLINE 639,92 TO 639,12
420 DRAWLINE 639,12 TO 71,12
430 DRAWLINE 71,108 TO 71,188
440 DRAWLINE 71,188 TO 639,188
450 DRAWLINE 639,188 TO 639,108
460 DRAWLINE 639,108 TO 71,108
470 POSITION 1,4                          'Opisz górną część
480 PRINT T$
490 RO = 12
500 Y = 92
510 FOR K = 1 TO 5                  'Podziel zakres wartości górnej części
520   POSITION RO,3                 'rysunku  w pionie na 5 części.
530   L = LT + R1 * K / 5
540   PRINT USING "####.#";L
550   DRAWLINE 70,Y TO 639,Y
560   RO = RO - 2
570   Y = Y - 16
580 NEXT K
590 POSITION 25,4                         'Opisz dolną część
600 PRINT B$
610 RO = 14
620 Y = 108
630 FOR K = 1 TO 5                  'Podziel zakres wartości dolnej części
640   POSITION RO,3                 'rysunku  w pionie na 5 części
650   L = LB + R2 * K / 5
660   PRINT USING "####.#";L
670   DRAWLINE 70,Y TO 639,Y
680   RO = RO + 2
690   Y = Y + 16
700 NEXT K
710      'Opisz części
720 POSITION 13,1
730 FOR K = 1 TO N
740   P = INT((72+(K-1)*D+D/2)/PC - LEN(L$(K)/2 + 0.5) + 1
750   PRINT TAB(P);L$(K);
760 NEXT K
770      '*************** Narysuj słupki *****************
780  X1 = 71 + D/6
790 FOR K = 1 TO N
800   YT = INT((HT - T(K)) * RT + 12.5)
810   YB = INT(B(K) * RB + 108.5)
820   FOR X = X1 TO X1 + D * 2/3
830     DRAWLINE X,YT TO X,92
840     DRAWLINE X,YB TO X,108
850   NEXT X
860   X1 = X1 + D
870 NEXT K
880 END
```



Rys. 12.5. Porównawczy wykres słupkowy ze słupkami dla jednego zbioru danych narysowanymi w górę i ze słupkami dla drugiego zbioru danych narysowanymi w dół — wynik działania programu 12.5



Rys. 12.6. Kumulacyjny wykres powierzchniowy utworzony przez program 12.6; dane z jednego zbioru są wykreślone jako wykres dolny, a dane z drugiego zbioru dodane do pierwszych są narysowane jako wykres górny

Krzywe można także porównywać rysując wiele zbiorów danych na tym samym wykresie. Program 12.6 służy do rysowania kumulacyjnego wykresu powierzchniowego. Na wynikowym wykresie (rys. 12.6) krzywa górna odpowiada sumie danych z obydwóch zbiorów, a różne cieniowanie umożliwia ich odróżnienie. Ten typ wykresu może jednak prowadzić do błędnych interpretacji, gdyż górna krzywa pokazuje sumę danych z dwóch zbiorów, nie zaś aktualne wartości.

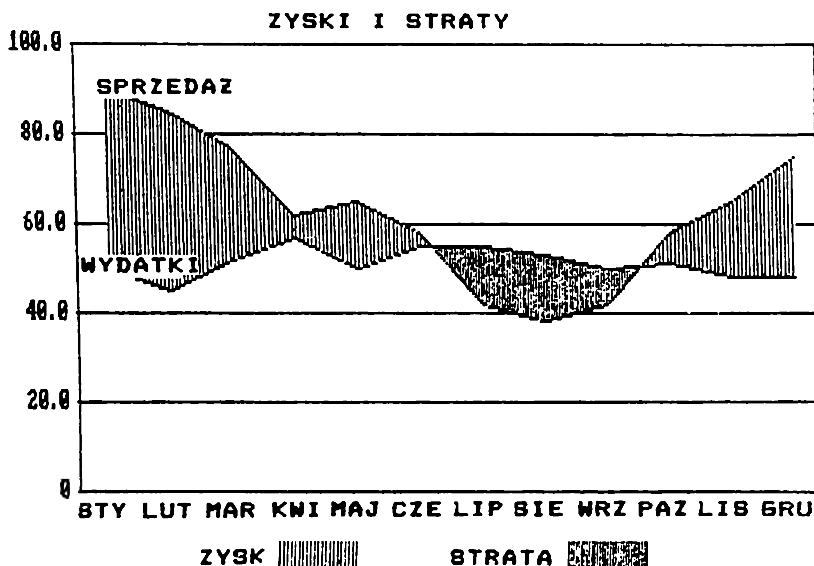
Program 12.6. Kumulacyjny wykres powierzchniowy dla dwóch zbiorów danych

```
10  'PROGRAM 12.6. Kumulacyjny wykres powierzchniowy dla dwóch zbiorów
20       'danych
25       'Program kreśli dolną krzywą dla jednego zbioru danych i górną
30       'krzywą jako sumę obydwu zbiorów. Program jest przystosowany
40       'do ekranu o wymiarach 640 x 200 pikseli.
50  '********************************************************************
60  CLEARSCREEN
70  DIM L$(20), S1(20), S2(20)
80  PC = 8                   'PC jest liczbą pikseli na znak w poziomie
90       '********************** Wprowadź dane ***********************
100 PRINT "Podaj tytuł wykresu";
110 INPUT T$
120 PRINT "Podaj nazwę pierwszego zbioru danych"
130 INPUT T1$
140 PRINT "Podaj nazwę drugiego zbioru danych"
150 INPUT T2$
160 PRINT "Podaj liczbę poziomych części"
170 INPUT N
180 D = INT(568 / N)
190 PRINT D
200 PRINT "Podaj nazwę i dwie wartości dla każdej części"
210 M2 = 0                   'M2 jest wartością maksymalną w tablicy S2
220 FOR K = 1 TO N
230   INPUT L$(K),S1(K),S2(K)
240   S2(K) = S1(K) + S2(K)
250   IF S2(K) <= M2 THEN 270
260   M2 = S2(K)
270   IF LEN(L$(K)) < D/PC THEN 310     'Czy nazwa się zmieści?
280   PRINT "Nazwa za długa. Maksymalna długość wynosi ";D/PC
290   PRINT "Podaj nową nazwę";
300   INPUT L$(K)
310 NEXT K
320 PRINT "Podaj wartości minimalną i maksymalną dla osi Y"
330 INPUT LO, HI
340 IF HI >= M2 THEN 380                'Czy HI jest dość duże?
350 PRINT "Wartość maksymalna za mała. Podaj nową wartość"
360 INPUT HI
370 GOTO 340
380 R = HI - LO                         'R jest zakresem wartości Y
390 RS = (172 - 12) / R       'RS służy do skalowania wykresu
400      '************ Narysuj osie i opisz je *******************
410 CLEARSCREEN
420 GRAPHICS
430 DRAWLINE 71,12 TO 71,172
440 DRAWLINE 71,172 TO 639,172
450 DRAWLINE 639,172 TO 639,12
460 DRAWLINE 639,12 TO 71,12
470 POSITION 1,1                        'Napisz tytuł rysunku
480 P = 40 - LEN(T$) / 2                'Ustaw tytuł w środku
490 PRINT TAB(P);T$
500 RO = 22
510 Y = 172
520 FOR K = 1 TO 5                 'Podziel zakresu wartości rysunku
530   POSITION RO,3                'w pionie  na 5 części
```

Program 12.6 (cd.)

```
540    L = LO + R * K / 5
550    PRINT USING "####.#";L
560    DRAWLINE 70,Y TO 639,Y
570    RO = RO - 4
590    Y = Y - 32
590 NEXT K
600       'Opisz częsci
610 POSITION 23,1
620 FOR K = 1 TO N
630    P = INT((71+(K-1)*D+D/2)/PC - LEN(L$(K)/2 + 0.5)) + 1
640    PRINT TAB(P);L$(K);
650 NEXT K
660       'Oznacz kodami kolorowymi zbiory danych
670 POSITION 25,28-LEN(T1$)
680 COLOR 2,0                          'Pierwszy zbiór w kolorze niebieskim
690 PRINT T1$;
700 FOR X = 228 TO 288
710    DRAWLINE X,190 TO X,199
720 NEXT X
730 DRAWLINE 228,190 TO 288,190
740 DRAWLINE 228,199 TO 288,199
750 POSITION 25,28-LEN(T2$)
760 COLOR 5,0                                  'Drugi zbiór w kolorze żółtym
770 PRINT T2$;
780 FOR X = 452 TO 512
790    DRAWLINE X,190 TO X,199
800 NEXT X
810 DRAWLINE 452,190 TO 512,190
820 DRAWLINE 452,199 TO 512,199
830       '************* Narysuj cieniowanie **********************
840 YO = 171
850 X1 = 71 + D/2     'Umieść pierwszy punkt po środku pierwszej części
860 Y1 = INT(HI -S2(1)) * RS + 12.5)
870 COLOR 5,0         'Zacieniuj obszar poniżej górnej krzywej na żółto
880 FOR K =2 TO N
890    X2 = X1 + D
900    Y2 = INT((HI - S2(K)) * RS + 12.5)
910    M = (Y2-Y1) / (X2-X1)
920    B = Y1 + 1 - M * X1
930    FOR X = X1 TO X2
940      Y = M * X + B
950    NEXT X
960    DRAWLINE X1,Y1 TO X2,Y2               'Zacieniuj pierwszy obszar
970    X1 = X2                               'Zapamiętaj aktualny punkt
980    Y1 = Y2
990 NEXT K
1000       'Rysuj dolną krzywą (pierwszy zbiór danych)
1010 X1 = 71 + D/2
1020 Y1 = INT(HI -S1(1)) * RS + 12.5)
1030 COLOR 2,0    'Zacieniuj obszar poniżej dolnej krzywej na niebiesko
1040 FOR K =2 TO N
1050    X2 = X1 + D
1060    Y2 = INT((HI - S1(K)) * RS + 12.5)
1070    M = (Y2-Y1) / (X2-X1)
1080    B = Y1 + 1 - M * X1
1090    FOR X = X1 TO X2
1100      Y = M * X + B
1110      DRAWLINE X,Y TO X,YO
1120    NEXT X
1130    DRAWLINE X1,Y1 TO X2,Y2            'Zacieniuj drugi obszar
1140    X1 = X2
1150    Y1 = Y2
1160 NEXT K
1170 END
```

W celu wyróżnienia obszarów między krzywymi można stosować różne wzory cieniowania (rys. 12.7). Ten wykres pasmowy został narysowany przez program 12.7. Z metody tej możemy korzystać do oceny różnic wielkości. W celu wyróżnienia wartości dodatnich (zyski) i ujemnych (straty) posłużono się różnymi wzorami cieniowania.



Rys. 12.7. Wykres pasmowy z różnymi wzorami cieniowania obszaru między krzywymi

Program 12.7. Wykres pasmowy z różnym cieniowaniem obszaru między krzywymi

```
10  'PROGRAM 12.7. Wykres pasmowy dwóch zbiorów danych
20  'Program rysuje wykresy dla dwóch zbiorów danych. Obszar, w którym
30  'krzywe się przecinają jest zacieniowany. Wymiary ekranu 640 x 200.
40  '**************************************************************
50  CLEARSCREEN
60  DIM L$(20), S1(20), S2(20)
70  PC = 8                        'PC jest liczbą pikseli na znak w poziomie
80       '*********************** Wprowadź dane ************************
90 PRINT "Podaj tytuł wykresu";
100 INPUT T$
110 PRINT "Podaj nazwę pierwszego zbioru danych"
120 INPUT T1$
130 PRINT "Podaj nazwę drugiego zbioru danych"
140 INPUT T2$
150 PRINT "Podaj nazwę obszaru, w którym dane pierwszego  zbioru"
155 PRINT " są większe niż dane drugiego zbioru"
160 INPUT T3$
170 PRINT "Podaj nazwę obszaru, w którym dane pierwszego  zbioru"
175 PRINT " są mniejsze niż dane drugiego zbioru"
180 INPUT T4$
```

Program 12.7 (cd.)

```
190 PRINT "Podaj liczbę poziomych części"
200 INPUT N
210 D = INT(568 / N)              'D jest liczbą pikseli na część
220 IF D2 <> INT(D/2) THEN D = INT(D/2+0.5)    'Niech D będzie parzyste
230 PRINT "Podaj nazwę i dwie wartości dla każdej części"
240 FOR K = 1 TO N
250   INPUT L$(K),S1(K),S2(K)
260   IF LEN(L$(K)) < D/PC THEN 300      'Czy nazwa się zmieści?
270   PRINT "Nazwa za długa. Maksymalna długość wynosi ";D/PC
280   PRINT "Podaj nową nazwę";
290   INPUT L$(K)
300 NEXT K
310 PRINT "Podaj wartości minimalną i maksymalną dla osi Y"
320 INPUT LO, HI
330 R = HI - LO                            'R jest zakresem wartości Y
340 RS = (172 - 12) / R           'RS służy do skalowania wykresu
350      '************ Narysuj osie i opisz je ******************
360 CLEARSCREEN
370 GRAPHICS
380 DRAWLINE 71,12 TO 71,172
390 DRAWLINE 71,172 TO 639,172
400 DRAWLINE 639,172 TO 639,12
410 DRAWLINE 639,12 TO 71,12
420 POSITION 1,1                           'Napisz tytuł rysunku
430 P = 40 - LEN(T$) / 2                   'Ustaw tytuł w środku
440 PRINT TAB(P);T$
450 RO = 22
460 Y = 172
470 FOR K = 1 TO 5                         'Podziel zakres wartości rysunku
480   POSITION RO,3                        'w pionie na 5 części
490   L = LO + R * K / 5
500   PRINT USING "####.#";L
510   DRAWLINE 70,Y TO 639,Y
520   RO = RO - 4
530   Y = Y - 32
540 NEXT K
550      'Opisz części
560 POSITION 23,1
570 FOR K = 1 TO N
580   P = INT((71+(K-1)*D+D/2)/PC - LEN(L$(K)/2 + 0.5) + 1
590   PRINT TAB(P);L$(K);
600 NEXT K
610      'Oznacz obszary kolorowymi kodami
620 POSITION 25,28-LEN(T3$)
630 PRINT T3$;
640 FOR X = 228 TO 288 STEP 3
650   DRAWLINE X,190 TO X,199
660 NEXT X
670 POSITION 25,28-LEN(T4$)
680 PRINT T4$;
690 FOR X = 452 TO 512 STEP 2
700   DRAWLINE X,190 TO X,199
710 NEXT X
720      '****************** Narysuj krzywe *********************
730 YO = 171
740 X1 = 71 + D/2     'Umieść pierwszy punkt po środku pierwszej części
750 Y1 = INT(HI -S1(1)) * RS + 12.5)
760 R1 = INT(Y1/PC + 0.5)        'X1,Y1 i X2,Y2 dotyczą pierwszej krzywej
770 C1 = INT(X1/PC + 0.5)
780 X3 = X1                      'X3,Y3 i X4,Y4 dotyczą drugiej krzywej
790 Y3 = INT(HI -S2(1)) * RS + 12.5)
800 R3 = INT(Y3/PC + 0.5)
810 C3 = INT(X3/PC + 0.5)
```

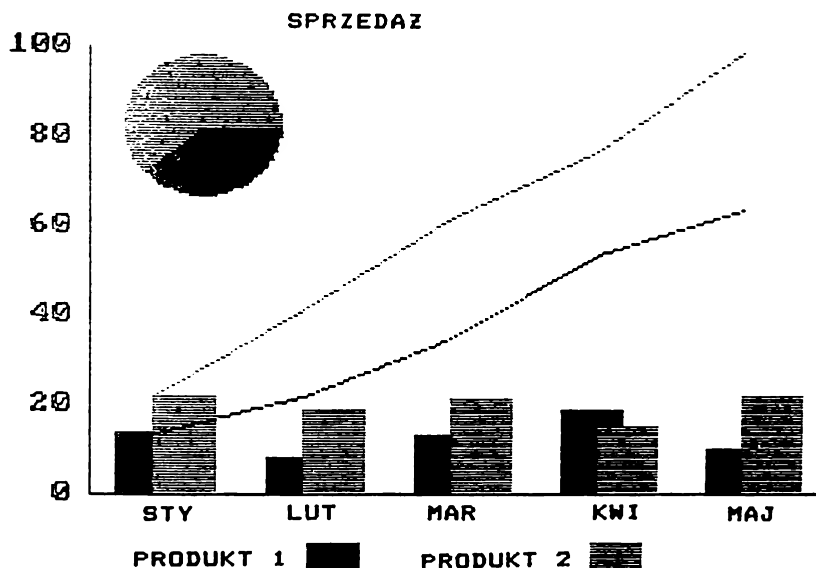
Program 12.7 (cd.)

```
820 FOR K =2 TO N
830   X2 = X1 + D
840   Y2 = INT((HI - S1(K)) * RS + 12.5)
850   M1 = (Y2-Y1) / (X2-X1) 'Nachylenie i przesunięcie pierwszej krzywej
860   B1 = Y1 - M1 * X1
870   X4 = X2
880   Y4 = INT((HI - S2(K)) * RS + 12.5)
890   M3 = (Y4-Y3) / (X4-X3) 'Nachylenie i przesunięcie drugiej krzywej
900   B3 = Y3 - M3 * X3
910   'Wzór cieniowania (rysowana co druga lub co trzecia linia) wynika
920   'z tego, która krzywa jest na górze. Rysuj co 2 linię, jeśli na
930   'górze jest krzywa 3-4, w przeciwnym przypadku co 3 linię.
940   IF Y3<=Y1 AND Y4<=Y2 THEN 1050  'Krzywa 3-4 jest wyżej niż krzywa 1-2
950   IF Y3>Y1 AND Y4>Y2 THEN 1070    'Krzywa 1-2 jest wyżej niż krzywa 3-4
960   'Krzywe przecinają się
970   XP = INT((B1-B3)/(M3-M1)+0.5)   'Znajdź wartość wsp. X p. przecięcia
980   IF Y3 > Y1 THEN 1020
990   B=X3: E=XP: I=2: GOSUB 1230  'Krzywa 3-4 na górze aż do przecięcia
1000  B=XP: E=X4: I=3: GOSUB 1230  'Od przec. do XP krzywa 1-2 na górze
1010  GOTO 1080
1020  B = X3: E = XP: I = 3: GOSUB 1230
1030  B = XP: E = X4: I = 2: GOSUB 1230
1040  GOTO 1080
1050  B = X3: E = X4: I = 2: GOSUB 1230
1060  GOTO 1080
1070  B = X3: E = X4: I = 3: GOSUB 1230
1080  DRAWLINE X1,Y1 TO X2,Y2             'Rysuj obydwie krzywe
1090  DRAWLINE X3,Y3 TO X4,Y4
1100   X1 = X2
1110   Y1 = Y2
1120   X3 = X4
1130   Y3 = Y4
1140 NEXT K
1150 POSITION R1,C1                       'Opisz początek krzywych
1160 PRINT T1$
1170 POSITION R3,C3
1180 PRINT T3$
1190 GOTO 1330
1200 '
1210 '****************** Zacieniuj obszary *******************
1220 COLOR 0,0                            'Wymaż poziome kreski podziału
1230 FOR X = B TO E
1240   YF = M1 * X + B1
1250   YS = M3 * X + B3
1260   DRAWLINE X,YF TO X,YS
1270   IF X/I<>INT(X/I) THEN 1300         'Czy linia jest wielokrotnością I?
1280   COLOR 1,0                          'Jeśli tak, zmień kolor i rysuj
1290   DRAWLINE X,YF TO X,YS
1300 NEXT X
1310 RETURN
1320 '*******************************************************
1330 END
```

12.3. Wykresy złożone

Jedna z użytecznych metod porównywania dwóch zbiorów danych polega na wykreśleniu ich w kilku postaciach na tym samym rysunku. Przykład rysunku (uzyskany za pomocą programu 12.8) zawierającego wykres słupkowy, wykres liniowy i wy-

kres kołowy, przedstawiono na rys. 12.8. Względne wielkości danych pokazano na wykresie słupkowym. Wykres liniowy prezentuje wielkości zsumowane (wykres kumulacyjny), a wykres kołowy procentowy udział zbiorów. W celu identyfikacji zbiorów zastosowano różne kolory.



Rys. 12.8. Wykres złożony utworzony za pomocą programu 12.8

Program 12.8. Wykres złożony: wykres słupkowy, liniowy i kołowy

```
10  'PROGRAM 12.8. Dwa zbiory danych przedstawione w różnych postaciach
20       'Program wprowadza dwa zbiory danych i rysuje ich wykres
30       'słupkowy, kołowy dla pokazania ich udziału procentowego oraz
40       'wykres kumulacyjny w funkcji czasu. W programie zakłada się,
50       'że ekran ma wymiary 320 x 200 pikseli.
60       '*************************************************************
70  CLEARSCREEN
80  DIM L$(5),S1(5), S2(5), C1(5), C2(5)
90  PC = 8                    'PC jest liczbą pikseli na znak w poziomie
100 XC = 73                   'XC,YC oznacza środek wykresu kołowego
110 YC = 40
120 RA = 30                   'RA jest promieniem wykresu kołowego
130 YA = 5/6                  'YA  dopasowuje do wybranej rozdzielczości
140      '********************* Wprowadź dane ***********************
150 PRINT "Podaj tytuł wykresu";
160 INPUT T$
170 PRINT "Podaj nazwę pierwszego zbioru danych"
180 INPUT T1$
190 PRINT "Podaj nazwę drugiego zbioru danych"
200 INPUT T2$
210 PRINT "Podaj liczbę części"
220 INPUT N
230 D = INT(290 / N)              'D jest liczbą pikseli na część
240 T1 = 0               'T1 i T2 podają wartości sumaryczne danych
```

Program 12.8 (cd.)

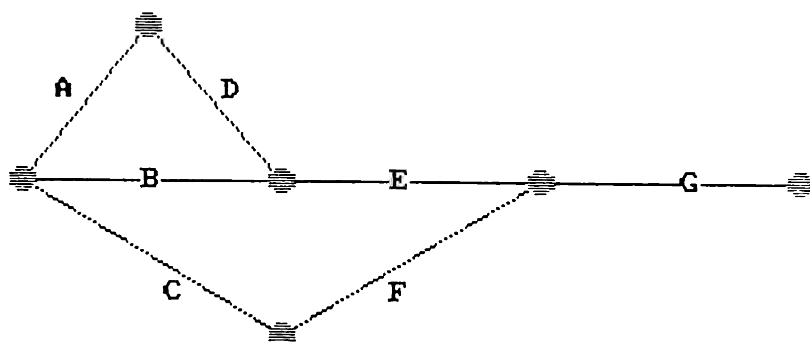
```
250 T2 = 0
260 PRINT "Podaj nazwę i dwie wartości dla każdej części"
270 FOR K = 1 TO N
280   INPUT L$(K),S1(K),S2(K)
290   T1 = T1 + S1(K)
300   C1(K) = T1
310   T2 = T2 + S2(K)
320   C2(K) = T2
330   IF LEN(L$(K)) < D/PC THEN 370     'Czy nazwa się zmieści?
340   PRINT "Nazwa za długa. Maksymalna długość wynosi ";D/PC
350   PRINT "Podaj nową nazwę";
360   INPUT L$(K)
370 NEXT K
380 PRINT "Podaj wartości minimalną i maksymalną dla osi Y"
390 INPUT LO, HI
400 IF HI >= T1 HI >= T2 THEN 440
410 PRINT "Wartość maksymalna za mała. Podaj nową wartość"
420 INPUT HI
430 GOTO 400
440 R = HI - LO                            'R jest zakresem wartości Y
450 RS = (172 - 12) / R          'RS służy do skalowania wykresu
460       '************ Narysuj osie i opisz je *******************
470 CLEARSCREEN
480 GRAPHICS
490 DRAWLINE 30,12 TO 30,172
500 DRAWLINE 30,172 TO 319,172
510 POSITION 1,1                           'Napisz tytuł rysunku
520 P = 20 - LEN(T$) / 2                   'Ustaw tytuł w środku
530 PRINT TAB(P);T$
540 RO = 22
550 Y = 172
560 FOR K = 0 TO 5                         'Podziel zakres wartości rysunku
570   POSITION RO, 1                       'w pionie na 5 części i opisz
580   L = LO + R * K / 5
590   PRINT USING "####.#";L
600   RO = RO - 4
610   Y = Y - 32
620 NEXT K
630       'Opisz części
640 POSITION 23,1
650 FOR K = 1 TO N
660   P = INT((30+(K-1)*D+D/2)/PC - LEN(L$(K))/2 + 0.5) + 1
670   PRINT TAB(P);L$(K);
680 NEXT K
690       'Oznacz kodami kolorowymi zbiory danych
700 POSITION 25,14-LEN(T1$)
710 COLOR 2,0                    'Pierwszy zbiór w kolorze niebieskim
720 PRINT T1$;
730 FOR X = 114 TO 134
740   DRAWLINE X,190 TO X,199
750 NEXT X
760 POSITION 25,28-LEN(T2$)
770 COLOR 4,0                    'Drugi zbiór w kolorze zielonym
780 PRINT T2$;
790 FOR X = 226 TO 246
800   DRAWLINE X,190 TO X,199
810 NEXT X
820       '
830       '****************** Rysuj słupki *************************
840 YO = 171
850 X1 = 30 + D/6
860 FOR K = 1 TO N
870   Y = INT((HI - S1(K)) * RS + 12.5)
```

Program 12.8 (cd.)

```
880    COLOR 2,0
890    FOR X = X1 TO X1 + D * 5/12
900      DRAWLINE X,Y TO X,Y0
910    NEXT X
920    X2 = X1 + D * 3/12                        'Drugi słupek przesunięty o 3/12
930    Y = INT((HI - S2(K)) * RS + 12.5)            'w stosunku do pierwszego
940    COLOR 4,0
950    FOR X = X2 TO X2 + D * 5/12
960      DRAWLINE X,Y TO X,Y0
970    NEXT X
980    X1 = X1 + D
990 NEXT K
1000     '
1010     '************************ Rysuj linie *****************************
1020 X = 30 + D * 5/12
1030 Y1 = INT((HI - C1(K)) * RS + 12.5)
1040 Y2 = INT((HI - C2(K)) * RS + 12.5)
1050 FOR K = 2 TO N
1060   Y3 = INT((HI - C1(K)) * RS + 12.5)
1070   Y4 = INT((HI - C2(K)) * RS + 12.5)
1080   COLOR 2,0
1090   DRAWLINE X,Y1 TO X+D,Y3
1100   COLOR 4,0
1110   DRAWLINE X,Y2 TO X+D,Y4
1120   X = X + D
1130   Y1 = Y3
1140   Y2 = Y4
1150 NEXT K
1160     '
1170     '******************** Rysuj wykres kołowy **********************
1180 CT = T1 + T2                  'CT jest sumą obydwu zbiorów
1190     'A1 jest ułamkiem okręgu odpowiadającym pierwszemu zbiorowi
1200 A1 = 360 * 3.14159 / 180 * T1 / CT
1210 DA = 1/RA/15              'Ustal mały krok aby ładnie zapełnić koło
1220 COLOR 2,0
1230 FOR A = DA TO A1 STEP DA
1240   X = XC + RA * COS(A)
1250   Y = YC + RA * SIN(A) * YA
1260   DRAWLINE XC,YC TO X,Y
1270 NEXT A
1280     'Zapełnij resztę koła kolorem odpow. drugiemu zbiorowi
1290 COLOR 4,0
1300 FOR A = A1 + DA TO 6.28318 STEP DA
1310   X = XC + RA * COS(A)
1320   Y = YC + RA * SIN(A) * YA
1330   DRAWLINE XC,YC TO X,Y
1340 NEXT A
1350 END
```

12.4. Wykresy do koordynacji projektów

Wykresy przedstawiające sieci działań (rys. 12.9) mogą być pomocne przy koordynacji i kontroli prac. Taki rodzaj wykresu jest szczególnie użyteczny przy koordynacji prac metodą PERT ze wspomaganiem komputerowym. Prace przedstawione na tym wykresie w postaci gałęzi grafu są uporządkowane od lewej do prawej strony zgodnie z czasem ich rozpoczęcia. Kółka służą do oznaczenia początku i końca pra-

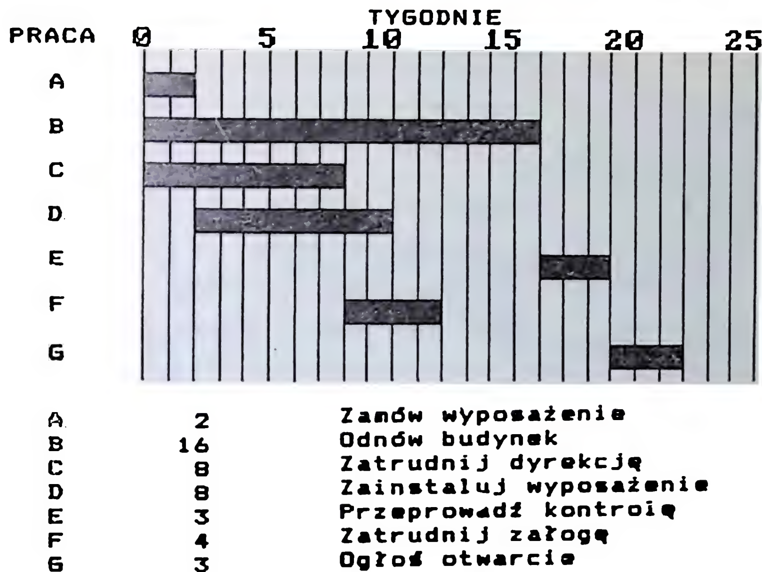


A	2	Zamów wyposażenie
B	16	Odnów budynek
C	8	Zatrudnij dyrekcję
D	8	Zainstaluj wyposażenie
E	3	Przeprowadź kontrolę
F	4	Zatrudnij załogę
G	3	Ogłoś otwarcie

Rys. 12.9. Sieć działań przedstawiająca kolejność różnych zadań w projekcie

cy. Kółko z lewej strony oznacza początek projektu, w którym prace A, B i C mogą być rozpoczęte jednocześnie. Praca D nie może być rozpoczęta przed końcem pracy A, a praca E rozpocznie się wówczas, gdy zostaną zakończone prace A, B i D.

Zadania projektowe można także przedstawić w postaci diagramu czasowego, pokazującego rzeczywiste daty rozpoczęcia i zakończenia prac. Diagram czasowy projektu zilustrowany na rys. 12.10 utworzono za pomocą programu 12.9. Ten diagram określa (w tygodniach) początek i koniec każdej pracy w projekcie. Prace są reprezentowane przez poziome paski o długościach proporcjonalnych do zaplanowanego czasu trwania pracy. Wykresy w postaci sieci działań i w postaci diagramu czasowego stosuje się przy planowaniu i kierowaniu projektami.



Rys. 12.10. Diagram czasowy, uzyskany za pomocą programu 12.9, pokazujący zależności czasowe różnych zadań w projekcie

Program 12.9. Diagram czasowy koordynacji prac

```
10  'PROGRAM 12.9. Diagram czasowy
20       'Zakładamy, że ekran ma wymiary 320 x 200 pikseli z 40 znakami
30       'w wierszu. Wykres mieści się w kolumnach od 60 do 300.
40       '
50       '************************************************************
60  CLEARSCREEN
70  GRAPHICS
80  READ TD$                      'Opis osi czasu
90  READ ET,LT                    'Początek i koniec osi czasowej
100 RT = LT - ET                  'RT jest zakresem czasu
110 RS = (300 - 60) / RT          'RS służy do skalowania
120 POSITION 1,20
130 PRINT TD$
140 POSITION 2,2
150 PRINT "PRACA";
160 CO = 6
170 X = 60
180 FOR K = 0 TO 5                'Opisz oś czasu co 1/5 zakresu
190   POSITION 2,CO
200   T = ET + RT * K/5
210   PRINT USING "###";T;
220   X = X + 48
230   CO = CO + 6
240 NEXT K
250 I = 240 / 25               'Rysuj 25 kolumn między pikselami 60 a 300
260 X = 60
270 FOR L = 1 TO 26               'W sumie 26 linii
280   DRAWLINE X,16 TO X,132
290   X = X + I
300 NEXT L
```

Program 12.9 (cd.)

```
310 R1 = 19                    'W wierszu 19 wypisz tablice
320 R2 = 4                     'W wierszu 4 opisz paski
330 FOR K =1 TO 7
340   READ C$,T$,S,D           'Kod, nazwa pracy, start i czas trwania
350   POSITION R1,4            'Tablica na dole
360   PRINT C$;
370   PRINT TAB(10);USING "##";D;
380   PRINT TAB(18);T$;
390   POSITION R2,4            'Rysuj paski na górze
400   PRINT C$;
410   X1 = S * RS + 60         'Znajdź początek paska
420   X2 = (S + D) * RS + 60   'Znajdź koniec paska
430   Y1 = (R2 - 1) * PR - 1   'Wartości Y różne dla każdego
440   Y2 = Y1 + 9              'paska
450   FOR Y = Y1 TO Y2         'Paski o wysokości 10 pikseli
460     DRAWLINE X1,Y TO X2,Y
470   NEXT Y
480   R1 = R1 + 1
490   R2 = R2 + 2
500 NEXT K
510     '*************************************************************
520 DATA TYGODNIE 0,25
530 DATA A,Zamów wyposażenie,0,2
540 DATA B,Odnów budynek,0,16
550 DATA C,Zatrudnij dyrekcję,0,8
560 DATA D,Zainstaluj wyposażenie,2,8
570 DATA E,Przeporwadź kontrolę,16,3
580 DATA F,Ztrudnij załogę,8,4
590 DATA G,Ogłoś otwarcie,19,3
600 END
```

13. Grafika w nauczaniu

Dostępność tanich mikrokomputerowych systemów graficznych daje ogromne możliwości edukacyjne od szkoły podstawowej poczynając a na wyższej uczelni kończąc. Programy graficzne mogą dobrze wspomagać pracę w klasie i być podstawą samodzielnych projektów w laboratorium. Programy wspomagające nauczanie można podzielić na trzy obszerne klasy: programy do powtarzania i ćwiczeń, programy uczące i sprawdzające oraz programy symulacyjne.

13.1. Programy do powtarzania i ćwiczeń

Przy korzystaniu z programów do powtarzania i ćwiczeń prezentujemy na ekranie pytania i żądamy na nie odpowiedzi. Mogą to być na przykład pytania na temat struktury zdania, języków obcych, osób i postaci historycznych, form artystycznych lub struktur geologicznych. Odpowiedzi mogą być wybierane z listy lub wpisywane na klawiaturze. Program do powtarzania i ćwiczeń ocenia odpowiedź wyświetlając proste zdania typu DOBRZE lub ŹLE, SPRÓBUJ JESZCZE RAZ. Zwykle celowe jest dysponowanie programem, który bardziej precyzyjnie określi błędy odpowiedzi. Przykładowo, program z chemii do ćwiczeń na temat tablicy pierwiastków Mendelejewa przy pytaniu o liczbę atomową przypadkowo wybranego pierwiastka mógłby błędną odpowiedź kwitować informacją, że podaną liczbę atomową ma pierwiastek taki a taki (jeżeli w ogóle istnieje). Programy bardziej rozbudowane mogą na błędną odpowiedź reagować szeregiem pytań naprowadzających lub wyświetlaniem obrazów i tekstu ułatwiających znalezienie poprawnej odpowiedzi.

W tego rodzaju programach istotnym elementem jest dobrze dobrana grafika. Program do ćwiczenia czytania i wymowy może rysować obraz obiektu na ekranie (samochód, drzewo, łódka, dom), a program dotyczący ekonomii może korzystać z wykresów i diagramów danych gospodarczych. W celu urozmaicenia

Program 13.1. Program ćwiczeń w dodawaniu

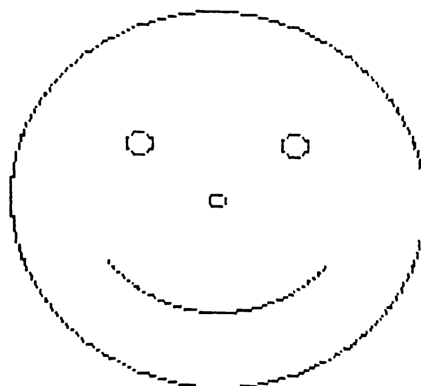
```
10  'PROGRAM 13.1. Ćwiczenia w arytmetyce
20      'Program - używając generatora liczb losowych - podaje ćwicze-
30      'nie arytmetyczne w postaci tekstowej (3 + 4 = ?) oraz w
40      'postaci kółek do policzenia, jeśli uczeń dał dwie błędne
50      'odpowiedzi. Jeśli nadal program otrzyma dwie błędne
60      'odpowiedzi, to przechodzi do następnego zadania. Pamięta
70      'się liczbę poprawnych odpowiedzi (R) uzyskanych w pierwszej
80      'próbie (gdy C = 1). Po 5 próbach, jeśli liczba R jest większa
90      'niż 3, pojawia się uśmiechnięta twarz i napis pochwalny.
100     'Po każdych 10 zadaniach pojawia się nowy obraz i nowy
110     'napis, jeśli liczba poprawnych odpowiedzi R jest większa
120     'niż 8.
130     '*****************************************' **************
140 CLEARSCREEN
150 GRAPHICS
160 PRINT "Halo! Jak Ci na imię?";
170 INPUT N$
180 PRINT "Dobrze, ";N$;" , zaczynamy!";
190 PRINT "Sprawdź, ile zadań rozwiążesz dobrze!"
200 FOR K = 1 TO 1000:NEXT K
210     '******************* Zadaj 10 zadań ************************
220 FOR P = 1 TO 10
230   C = 0                     'C jest liczbą odpowiedzi na zadanie
240   J = INT(RND(1) * 9 + 0.5)
250   K = INT(RND(1) * 9 + 0.5)
260   CLEARSCREEN
270   POSITION 16,1
280   PRINT J;"+";K;"=";
290   INPUT A
300   C = C + 1
310   IF A = J + K THEN 490      'Poprawna odpowiedź?
320   IF C <> 1 THEN 380
330   POSITION 18,5              'Pierwsza odpowiedź. Powtórz
340   PRINT N$;" ,to nie jest dobra odpowiedź";
350   PRINT "         Spróbuj jeszcze raz"
360   FOR F = 1 TO 1000:NEXT F
370   GOTO 260
380   IF C <> 2 THEN 410
390   GOSUB 610                  'Dwukrotna błędna odpowiedź. Rysuj kółka
400   GOTO 270
410   IF C <> 3 THEN 460  'Trzecia błędna odpowiedź. Jeszcze jedna próba
420   GOSUB 610
430   POSITION 4,1
440   PRINT "Spróbuj jeszcze raz!";
450   GOTO 270
460   POSITION 20,1                     'Przejdź do następnego zadania
470   PRINT "Spróbujmy inne zadanie";
480   FOR F = 1 TO 1000:NEXT F
490   IF C = 1 THEN R = R + 1           'Pierwsza odpowiedź dobra
500   IF P=5 AND R>3 THEN GOSUB 790     'Narysuj uśmiechniętą twarz
510   IF P=10 AND R>8 THEN GOSUB 960    'Narysuj balony
520 NEXT P
530     '
540 CLEARSCREEN
550 POSITION 10,19
560 PRINT "Do widzenia!"
570 POSITION 12,20-LEN(N$)/2
580 PRINT N$
590 GOTO 1260
600     '
610     '************** Rysuj pomocnicze kółka ********************
620 CLEARSCREEN
630 Y = 76
```

Program 13.1 (cd.)

```
640 X = [illegible]
650 FOR W = 1 TO J
660   CIRCLEPLOT X,Y,5
670   X = X + 16
680 NEXT W
690 POSITION 10,X/8+1
700 PRINT "+";
710 X = X + 20
720 FOR W = 1 TO J
730   CIRCLEPLOT X,Y,5
740   X = X + 16
750 NEXT W
760 RETURN
770      '
780      '****************** Uśmiechnięta twarz ********************
790 CLEARSCREEN
800 POSITION 25,4
810 PRINT "Jak do tej pory - zupełnie dobrze, ";N$
820 CIRCLEPLOT 160,100,80
830 CIRCLEPLOT 130,80,5
840 CIRCLEPLOT 190,80,5
850 CIRCLEPLOT 160,100,3
860 FOR A = 0.79 TO 2.36 STEP 1/60      'Rysuj usta
870   XP = 160 + 60 * COS(A)
880   YP = 80 + 60 * SIN(A)
890   POINTPLOT XP,YP
900 NEXT A
910 FOR F = 1 TO 1500:NEXT F
920 COLOR 0,0
930 RETURN
940      '
950      '******************** Balony *********************  '
960 CLEARSCREEN
970 POSITION 4,17
980 PRINT N$
990 POSITION 6,18
1000 '
1010 '
1020 PRINT "Wspaniale!";
1030 CIRCLEPLOT 60,50,40
1040 CIRCLEPLOT 60,88,5
1050 DRAWLINE 60,91 TO 60,170          'Dorysuj sznurek do balonu
1060 CIRCLEPLOT 280,50,30
1070 CIRCLEPLOT 280,78,3
1080 DRAWLINE 280,79 TO 280,180
1090 CIRCLEPLOT 140,130,35
1100 CIRCLEPLOT 140,163,3
1110 DRAWLINE 140,164 TO 140,199
1120 FOR Y = 2 TO 199 STEP 5           'Rysuj konfetti
1130   X = RND(1) * 319
1140   CIRCLEPLOT X,Y,1
1150   FOR F = 1 TO 50:NEXT F
1160 NEXT Y
1170 X = 240: Y = 20: A = 0: R = 20: DA = 1/R/0.5
1180 FOR Y = 20 TO 199                 'Rysuj spiralę
1190   XP = X + R * COS(A)
1200   YP = Y + R * SIN(A)
1210   POINTPLOT XP,YP
1220   A = A + DA
1230 NEXT Y
1240 RETURN
1250 '*****************************************************************
1260 END
```

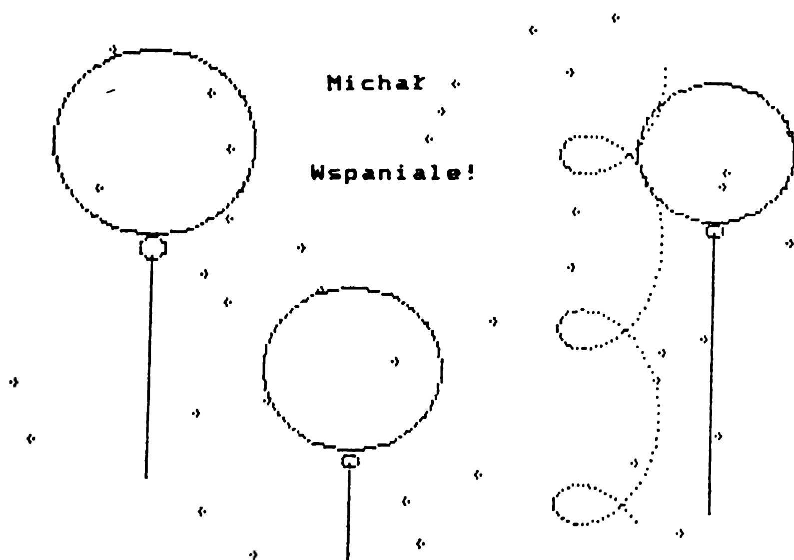
4 + 5 = ?

(a)



Jak do tej pory – zupełnie dobrze, Michał

(b)



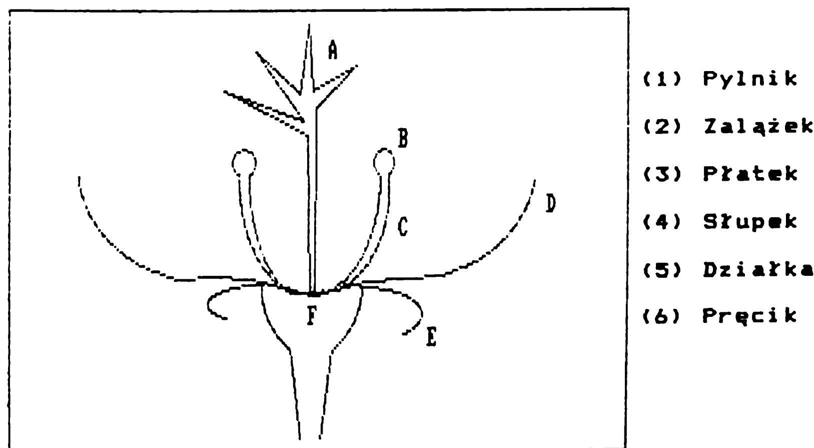
(c)

Rys. 13.1. Wyniki programu 13.1 umożliwiające ćwiczenie w dodawaniu przedstawiające: a) zachętę, gdy odpowiedź była błędna; b) uśmiechniętą twarz, gdy odpowiedź była dobra; c) końcową pozytywną ocenę z balonami i serpentynami

i uczynienia ćwiczenia bardziej interesującym możemy prezentację i oceny ilustrować rysunkami. Na przykład w programie 13.1, będącym ćwiczeniem w dodawaniu, korzysta się z grafiki zarówno w celu urozmaicenia wykładu, jak i do wizualnej pomocy dla ucznia. Na rysunku 13.1 pokazano przykładowy rysunek na ekranie przy pytaniu o sumę 4 + 5. W celu dodatkowego urozmaicenia możemy wybierać różne rysunki twarzy i tekstu oceny (DOBRZE, ZNAKOMICIE, NIE, FATALNIE). Podobnie elementy do sumowania mogą być za każdym razem inne.

13.2. Programy liczące i sprawdzające

W programach uczących i sprawdzających korzystamy z metod konwersacyjnych. Programy uczące mogą zawierać instrukcje podające materiał do przeczytania (książki, artykuły, filmy do obejrzenia) i studiowania w rytmie określonym przez uczącego się, a także mogą zawierać testy na temat zadanego materiału. Programy sprawdzające mogą prowadzić konwersację przez zadawanie pytań, ale także przez



Napisz numer części B z listy
po prawej stronie :

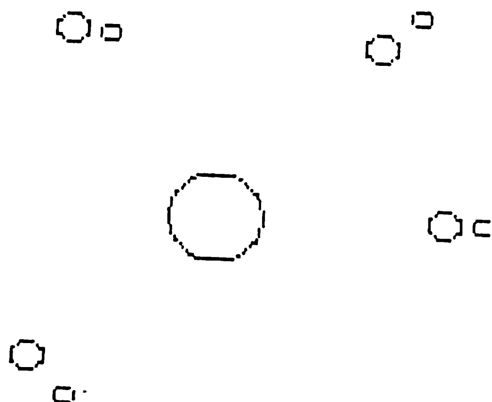
Rys. 13.2. Przykład rysunku z programu do samodzielnego studiowania

podawanie odpowiedzi. Na przykład program sprawdzający wspomaga nauczanie lekarzy przez odpowiadanie na pytania lekarza tak, jakby to robił żywy pacjent. Odpowiedzi te mogą być użyte do stawiania diagnozy. Te obydwa rodzaje programów korzystają z grafiki w taki sam mniej więcej sposób jak programy do powtarzania i ćwiczeń.

W przypadku programów do samodzielnej nauki możemy na przykład tworzyć obrazy jak na rys. 13.2, na którym pokazano części kwiatu oraz wykaz nazw. Program może tworzyć wiele takich rysunków, ocenić odpowiedzi i wydać całościową ocenę. Taki program może być stosowany do egzaminowania lub sprawdzania wiedzy.

13.3. Programy symulacyjne

Z programów symulacyjnych korzystamy zwykle do demonstrowania zachowania różnych systemów. Model rzeczywistego systemu fizycznego lub biologicznego (albo model systemu hipotetycznego) może służyć jako podstawa wykładu lub być częścią programu do samodzielnej nauki. Samodzielne programy są szczególnie przydatne przy studiowaniu systemów o wielu parametrach lub systemów, których nie możemy obserwować w rzeczywistości. Przykładami mogą być systemy geopolityczne, struktury atomowe lub cząstkowe, relatywistyczny ruch obiektów. Umożliwia to zmianę parametrów systemu i obserwację jego zachowania na ekranie.



Rys. 13.3. Symulacja ruchu Ziemi i Księżyca wokół Słońca. Pokazano kilka położeń Ziemi i Księżyca

Program 13.2 tworzy animowany model systemu słonecznego i demonstruje złożone trajektorie elementów tego systemu. Na rysunku 13.3 przedstawiono kilka faz ruchu Ziemi i Księżyca obracających się wokół Słońca. Program 13.2 produkuje naraz tylko jedno takie położenie, wymazując poprzednie. Program pokazuje ruch po kole, nie po elipsie, a obiekty nie zachowują skali.

Programy symulacyjne stosuje się jako gry edukacyjne. Program strzelania pokazujący trajektorię pocisku przy różnych kątach podniesienia lufy może ułatwić zrozumienie zasad ruchu i umożliwić przyswojenie zależności między tymi wielkościami.

Program 13.2. Symulacja: modelowanie systemu słonecznego z obracającą się Ziemią i Księżycem

```
10 'PROGRAM 13.2. System słoneczny
20     'Program ilustruje ruchy ciał niebieskich w systemie słone-
30     'cznym. Księżyc obraca się wokół Ziemi 12 razy na każdy obrót
40     'Ziemi wokół Słońca. Wybrano orbitę Księżyca (RM) równą 1/6
50     'orbity Ziemi (RE), tak aby szybkość kątowa Księżyca (1/RM)
60     'była 6 razy większa. Przyjmując krok ruchu równy 1/RM i 1/RE
70     'otrzymujemy 6-krotnie szybszy ruch Księżyca po orbicie niż
80     'ruch Ziemi. Nowe położenie Księżyca jest rysowane na ekranie
90     '2 razy częściej niż położenie Ziemi, tak że wydaje się,
100    ' że Księżyc porusza się dwa razy szybciej niż Ziemia.
110    '**************************************************************
120 CLEARSCREEN
130 XS = 128        'Słońce w środku ekranu o wymiarach 256 x 192 pikseli
140 YS = 96
150 RE = 72                    'RE jest promieniem orbity Ziemi
160 RM = 12                    'RM jest promieniem orbity Księżyca
170 S = 15                     'S jest promieniem Słońca
180 E = 5                      'E jest promieniem Ziemi
190 M = 3                      'M jest promieniem Księżyca
200 GRAPHICS
210 CIRCLEPLOT XS,YS,S         'Rysuj Słońce
220 MO = 1/RM             'Przyrost kątowy dla Księżyca wynosi 1/RM
230 FOR EO = 1/RE TO 6.28318 STEP 1/RE           'a dla Ziemi 1/RE
240   COLOR 0,0
250   CIRCLEPLOT XE,YE,E       'Wymaż Ziemię
260   XE = XS + RE * COS(EO)
270   YE = YS + RE * SIN(EO)
280   COLOR 1,0
290   CIRCLEPLOT XE,YE,E       'Rysuj Ziemię
300   COLOR 0,0
310   CIRCLEPLOT XM,YM,E       'Wymaż Księżyc
320   XM = XE + RM * COS(MO)
330   YM = YE + RM * SIN(MO)
340   COLOR 1,0
350   CIRCLEPLOT XM,YM,E       'Rysuj Księżyc
360   MO = MO + 1/RM
370   COLOR 0,0
380   CIRCLEPLOT XM,YM,E       'Wymaż Księżyc
390   XM = XE + RM * COS(MO)
400   YM = YE + RM * SIN(MO)
410   COLOR 1,0
420   CIRCLEPLOT XM,YM,E       'Rysuj Księżyc
430   MO = MO + 1/RM
440 NEXT EO
450 GOTO 220
460 END
```

13.4. Programy towarzyszące procesowi nauczania

Do wspomagania nauczania są potrzebne zarówno programy graficzne, jak i programy „obsługowe". Programy mogą służyć do obsługi bazy danych o uczniach i ocenach oraz do obliczania danych statystycznych. Umożliwia to tworzenie wykresów rozkładu stopni jednego egzaminu, jednego przedmiotu, jednego przedmiotu na przestrzeni lat lub wielu przedmiotów w dowolnym przedziale czasowym.

14. Grafika w domu

Jako końcowy przykład zastosowań grafiki przedstawimy jej zastosowania dla celów domowych i osobistych. Omówimy programy graficzne rozrywkowe, edukacyjne i użytkowe.

14.1. Grafika w prowadzeniu domu

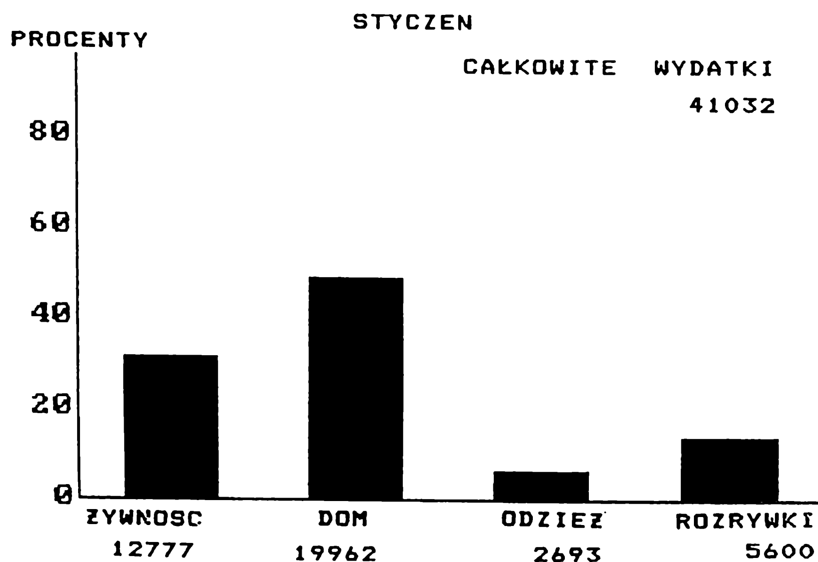
Istnieje wiele zastosowań grafiki w zakresie finansów, które mogą być użyteczne w domu. Możemy ilustrować graficznie różne wydatki, spłaty pożyczek, analizować dochody lub potencjalne oszczędności, jakie można by uzyskać dzięki wprowadzeniu urządzeń oszczędzających energię. Program 14.1 prezentuje przykład obliczania budżetu domowego. W programie uwzględniono tylko 4 kategorie wydatków (rys. 14.1). Dla każdej kategorii wydatków rysuje się słupek pokazujący ich procentowy udział w ogólnych wydatkach. Można wykreślić wiele rodzajów takich diagramów. Dane dotyczące wydatków tygodniowych lub miesięcznych mogą być wpisane do pliku. Wtedy można opracować program porównywania wydatków w kolejnych okresach.

Program 14.1. Budżet domowy

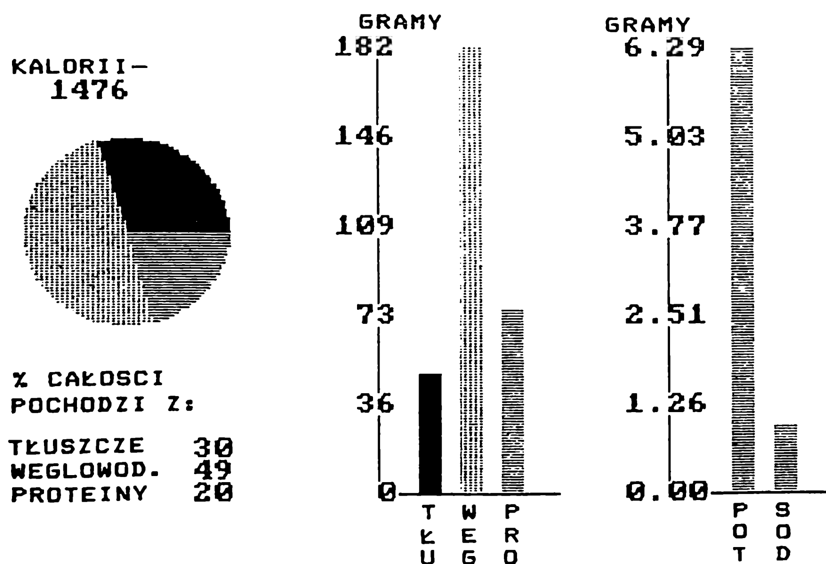
```
10  'PROGRAM 14.1. Budżet domowy
20       'Program rysuje wykres słupkowy wydatków domowych. Słupki
30       'pokazują procentowy udział całkowitych wydatków miesięcznych.
40       'Wydatki drukowane są w poszczególnych kategoriach:
50       'żywność, odzież, czynsz i świadczenia oraz rozrywki. Przyjęto,
60       'że wymiary ekranu wynoszą 320 x 200 pikseli.
70       '***************** Wprowadź dane ****************************
80  CLEARSCREEN
90  D = INT(290 / 4)             'Cztery kategorie wydatków
100 RS = (172 - 12)             'Słupki w tych liniach
110 T = 0                       'Całkowite wydatki
120 PRINT "Podaj nazwę miesiąca"
```

Program 14.1 (cd.)

```
130 INPUT M$
140 PRINT
150 PRINT "F - ZYWNOSC H - CZYNSZ i DOM C - ODZIEZ R - ROZRYWKI"
160 PRINT
170 PRINT "Podaj wydatki i kod kategorii. Napisz 0,K aby zakończyć"
180 INPUT E,C$
190 IF E = 0 AND C$ = "K" THEN 300
200 IF C$ = "F" OR C$ = "H" OR C$ = "C" OR C$ = "R" THEN 230
210 PRINT "Niewłaściwy kod wydatków"
220 GOTO 180
230 T = T + E
240       'Dodaj wydatki w każdej kategorii
250 IF C$ = "F" THEN FT = FT + E
260 IF C$ = "H" THEN HT = HT + E
270 IF C$ = "C" THEN CT = CT + E
280 IF C$ = "R" THEN RT = RT + E
290 GOTO 180
300       '************** Narysuj osie i opisz je ****************
310 CLEARSCREEN
320 GRAPHICS
330 DRAWLINE 25,12 TO 25,172
340 DRAWLINE 25,172 TO 315,172
350 POSITION 1,1                  'Pisz nazwę miesiąca
360 P = 20 - LEN(M$) / 2          'Ustaw nazwę na środku
370 PRINT TAB(P);M$
380 RO = 22
390 Y = 172
400 FOR K = 0 TO 4                'Opisz oś pionową co 1/5 zakresu
410   POSITION RO,1               'wartości
420   L = LO + 100 * K /5
430   PRINT USING "###";L
440   RO = RO - 4
450   Y = Y - 32
460 NEXT K
470 POSITION 2,1
480 PRINT "PROCENTY";
490       'Opisz części
500 POSITION 23,1
510 PRINT "    ZYWNOSC   DOM    ODZIEZ    ROZRYWKI";
520       '******************** Rysuj słupki ********************
530 YO = 171
540 X1 = 25 + D/4
550 Y = INT((1-FT/T)*RS+12.5)    'Jaki % całości przypada na każdą
560 GOSUB 700                     'kategorię? Odejmij od 1 (wtedy słupki
570 Y = INT((1-HT/T)*RS+12.5)    'będą rysowane w górę) i pomnóż przez
580 GOSUB 700                     'zakres pikseli przeznaczonych na
590 Y = INT((1-CT/T)*RS+12.5)    'słupki
600 GOSUB 700
610 Y = INT((1-RT/T)*RS+12.5)
620 GOSUB 700
630 POSITION 24,1
640 PRINT USNG " ####.##   ####.##  ####.##  ####.##";FT;HT;CT;RT;
650 POSITION 3,24
660 PRINT "CAŁKOWITE WYDATKI";
670 POSITION 4,30
680 PRINT USING "####.##";T
690 GOTO 770
700       '********************** Rysuj słupki ********************
710 FOR X = X1 TO X1 + D/2
720   DRAWLINE X,Y TO X,YO
730 NEXT X
740 X1 = X1 + D
750 RETURN
760       '*******************************************************
770 END
```



Rys. 14.1. Wykres budżetu domowego



Rys. 14.2. Diagram dietetyczny

Wykresy i diagramy stosuje się także do innych celów domowych. Na rysunku 14.2 przedstawiono wykres dietetyczny dla jednego dnia. Wykres uzyskano za pomocą programu 14.2, który oblicza liczbę spożytych kalorii i pokazuje procentowy udział kalorii zawartych w proteinach, węglowodanach i tłuszczach naszej diety. Dane dzienne możemy zapamiętać w pliku dyskowym i tak zmodyfikować program 14.2, aby uzyskać porównania w długich okresach. W takim przypadku celowe jest zwiększenie liczby danych na temat innych potraw i ich składu.

Program 14.2. Analiza dietetyczna: udział różnych potraw w liczbie spożytych kalorii

```
10  'PROGRAM 14.2. Analiza dietetyczna
20       'Program odczytuje nazwy potraw i ich skład z instrukcji DATA
30       'do tablic. Program odczytuje wagę (w gramach) tłuszczu,
40       'węglowodanów, protein, sodu i potasu dla potraw o średniej
50       'wadze. My wprowadzamy nazwy potraw dziś zjedzonych a program
60       'oblicza wagi składników tych potraw. Całkowita waga tłuszczów,
70       'węglowodanów itp. jest zamieniana na liczbę spożytych kalorii.
80       'Następnie wyświetla się wykres słupkowy pokazujący udział
90       'poszczególnych składników w całkowitej ilości spożytych
100      'kalorii. Rysuje się ponadto dwa wykresy słupkowe - jeden
110      'pokazuje ilości spożytych tłuszczów, węglowodanów i protein;
120      'drugi - ilości sodu i potasu. Przyjęto, że wymiary ekranu
130      'wynoszą 320 x 200 pikseli. Słupki są skalowane do zakresu
140      'linii od 12 do 172.
150      '**********************************************************
160 CLEARSCREEN
170 DIM F$(10),FA(10),CA(10),PR(10),PO(10),SO(10)
180 Y = 5/6
190 FOR K = 1 TO 10
200   READ F$(K),FA(K),CA(K),PR(K),PO(K),SO(K)
210 NEXT K
220      '************* Wprowadź opis wykresu i dane **************
230 PRINT "Podaj nazwę potrawy. Napisz K aby skończyć"
240 INPUT F1$
250 IF F1$ = "K" THEN 370
260 FOR K = 1 TO 10            'Znajdź nazwę potrawy w tablicy
270   IF F1$ K$(K) THEN 310
280 NEXT K
290 PRINT "Nieznana potrawa. Podaj następną nazwę potrawy"
300 GOTO 240
310 FAT = FAT + FA(K)          'Dodaj wszystkie składniki
320 CAT = CAT + CA(K)
330 PRT = PRT + PO(K)
340 POT = POT + PO(K)
350 SOT = SOT + SO(K)
360 GOTO 240
370      '************ Narysuj osie i opisz je ****************
380 CLEARSCREEN
390 GRAPHICS
400 H1 = PRT           'Znajdź wartość maks. dla pierwszego  słupka
410 IF FAT > H1 THEN H1 = FAT
420 IF CAT > H1 THEN H1 = CAT
430 R1 = (172 - 12) /H1    'R1 służy do skalowania pierwszego  słupka
440 H2 = POT               'Znajdź wartość maks. dla drugiego słupka
450 IF SOT > H2 THEN H2 = SOT
460 R2 = (172 - 12) /H2        'R2 służy do skalowania drugiego słupka
470 DRAWLINE 256,12 TO 256,172
480 DRAWLINE 256,172 TO 319,172
490 DRAWLINE 144,12 TO 144,172
500 DRAWLINE 144,172 TO 214,172
510 RO = 22
```

Program 14.2 (cd.)

```
520 Y = 172
530 FOR  K= 0 TO 5                'Opisz oś Y co 1/5 zakresu wartości
540   POSITION RO,17
550   L = H1 * K / 5
560   PRINT USING "###";L;
570   POSITION RO,31
580   L = H2 * K / 5
590   PRINT USING "#.##";L;
600   RO = RO -4
610   Y = Y- 32
620 NEXT K
630 POSITION 1,18
640 PRINT "GRAMY     GRAMY";
650      'Opisz części
660 POSITION 23,21
670 PRINT "T W P        P S";
680 POSITION 24,21
690 PRINT "Ł E R        O O";
700 POSITION 25,21
710 PRINT "U G O        T D";
720      '
730      '****************** Rysuj słupki ************************
740 YO = 171
750 X1 = 160
760 Y = INT((H1 - FAT) * R1 + 12.5)
770 COLOR 2,0                     'Niebieski
780 GOSUB 910                     'Narysuj słupek
790 Y = INT((H1 - CAT) * R1 + 12.5)
800 COLOR 5,0                     'Żółty
810 GOSUB 910                     'Narysuj słupek
820 Y = INT((H1 - PRT) * R1 + 12.5)
830 COLOR 4,0                     'Zielony
840 GOSUB 910                     'Narysuj słupek
850 X1 = 280                      'Przejdź do następnego wykresu
860 Y = INT((H2 - SOT) * R2 + 12.5)
870 GOSUB 910                     'Narysuj słupek
880 Y = INT((H2 - POT) * R2 + 12.5)
830 GOSUB 910                     'Narysuj słupek
900 GOTO 980
910      'Rysuj słupki
920 FOR X = X1 + 8
930   DRAWLINE X,Y TO X,YO
940 NEXT X
950 X1 = X1 + 16                  'Przejdź do następnego słupka
960 RETURN
970      '
980      '************* Rysuj wykres kołowy ****************
990  XC = 45
1000 YC = 78
1010 R = 40
1020 FAC = FAT * 9                'Oblicz liczbę kalorii
1030 PRC = PRT * 4
1040 CAC = CAT * 4
1050 CAL = FAC + CAC + PRC        'Całkowita ilość kalorii
1060 POSITION 3,1
1070 PRINT "KALORII - ";
1080 POSITION 4,3
1090 PRINT USING "###";CAL;
1100 POSITION 17,1
1110 PRINT "% CAŁOSCI";
1120 POSITION 18,1
1130 PRINT "POCHODZI Z";
1140 S = 0
1150 B = 0
```

Program 14.2 (cd.)

```
1160 S = S + FAC
1170 COLOR 2,0                  'Niebieski
1180 GOSUB 1320                 'Rysuj wycinek dla tłuszczów
1190 S = S + CAC
1200 COLOR 5,0                  'Żółty
1210 GOSUB 1320                 'Rysuj wycinek dla węglowodanów
1220 S = S + PRC
1230 COLOR 4,0                  'Zielony
1240 GOSUB 1320                 'Rysuj wycinek dla protein
1250 POSITION 20,1
1260 PRINT "TŁUSZCZE";TAB(12);USING"##";FAC/CAL*100;
1270 POSITION 21,1
1280 PRINT "WEGLOWOD.";TAB(12);USING"##";CAC/CAL*100;
1290 POSITION 22,1
1300 PRINT "PROTEINY";TAB(12);USING"##";PRC/CAL*100;
1310 GOTO 1540
1320     '************ Narysuj i zapełnij słupki *****************
1330 DA = 1/R/50          'Ustal mały krok aby ładnie zapełnić wykres
1340 A = 6.28318 * S /CAL          'A jest kątem końca wycinka
1350 FOR A1 = B TO A STEP DA    'B jest kątem ostatniego wycinka
1360   XP = XC + R * COS(A1)
1370   YP = YC + R * SIN(A1) * YA
1380   DRAWLINE XP,YP TO XC,YC
1390 NEXT A1
1400 B = A
1410 RETURN
1420     '*********************************************************
1420 DATA MLEKO,12,9,9,0.122,0.351
1430 DATA BEKON,1,8,5,0.163,0.038
1450 DATA DORSZ,5,6,20,0.177,0.348
1460 DATA TUNCZYK,0,18,21,0.688,0.259
1470 DATA JAJKA,0,6,7,0.066,0.070
1480 DATA SZPINAK,3,0,2,0.040,0.259
1490 DATA PŁATKI,1.4,26.32,4.48,0,0.231
1500 DATA DAKTYLE,0.89,129,3.9,0.002,1.15
1510 DATA WATROBA,0.6,0.3,1.5,0.022,3.04
1520 DATA SOS,5.2,10.4,6.4,0.001,0.45
1530     '*********************************************************
1540 END
```

Program 14.3 rysuje wykres biorytmów przedstawiony na rys. 14.3. Wykres prezentuje zmiany teoretycznej aktywności fizycznej, emocjonalnej i intelektualnej człowieka. Przyjęto, że okres aktywności fizycznej wynosi 23 dni, aktywności emocjonalnej 28 dni, a intelektualnej 33 dni.

Program 14.3. Biorytmy

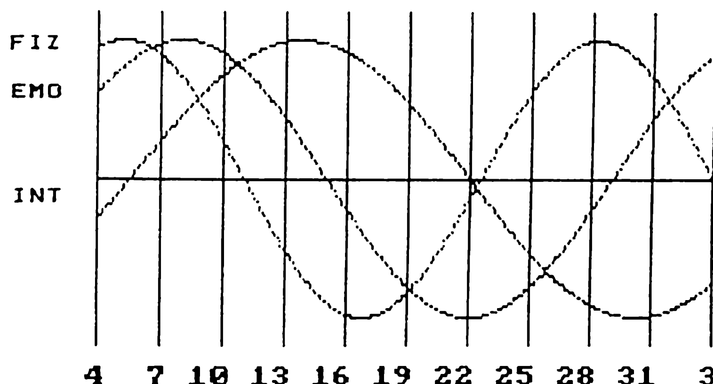
```
10  'PROGRAM 14.3. Biorytmy
20      'Program - po podaniu daty urodzin i początku biorytmu - tworzy
30      'wykres biorytmów (z krzywymi dla cykli emocjonalnych, fizycz-
40      'nych i intelektualnych) na następne 30 dni. W pobliżu krzywych
50      'znajdują się opisujące je etykiety. Etykiety umieszczano
60      'z lewej strony krzywych - o ile nie zaczynają się one zbyt
70      'blisko siebie: wtedy jedną z etykiet umieszcza się po prawej
80      'stronie wykresu. Przy sporządzaniu wykresu założono, że
90      'ekran ma wymiary 320 x 200 pikseli.
100     '*************************************************************
110 CLEARSCREEN
120 DIM M$(12), DM(12), AD(13)
```

Program 14.3 (cd.)

```
130 FOR K = 1 TO 12                'Odczytaj nazwę miesiąca, liczbę dni
140   READ M$(K),DM(K),AD(K)       'w miesiącu i sumę dni
150 NEXT K
160 PR = 8                         'PR jest liczbą pikseli na znak w pionie
170 PI = 3.14159
180 H = 50                         'H jest wysokością krzywych
190 RR = (284 - 44)/30             'RR jest współczynnikiem skalowania
200      'Wykres dla 30 dni zmieści się w kolumnach od 44 do 284
210      '****************** Wpowadź dane ************************
220 PRINT "Podaj datę urodzin (dzień, miesiąc, rok)"
230 INPUT DB,MB,YB
240 PRINT "Podaj datę początku wykresu (dzień, miesiąc, rok)"
250 INPUT DC,MC,YC
260      'Oblicz czas między datą urodzin a początkiem wykresu
270 T = (YC-YB)*365.25 + (AD(MC)-AD(MB))+(DC-DB)
280      'Oblicz przesunięcie dla każdej krzywej
290 PD = 2 * PI * (T / 23 - INT(T / 23))
300 ED = 2 * PI * (T / 28 - INT(T / 28))
310 ID = 2 * PI * (T / 33 - INT(T / 33))
320      'Oblicz częstotliwość dla każdej krzywej
330 PW = 2 * PI / 23
340 EW = 2 * PI / 28
350 IW = 2 * PI / 33
360      '
370      '********************** Rysuj osie ************************
380 CLEARSCREEN
390 GRAPHICS
400 POSITION 1,15
410 PRINT "BIORYTMY"
420 POSITION 3,20-(13+LEN(M$(MC)))/2
430 PRINT "POCZATEK: ";M$(MC);YC;
440 D = DC
450 X = 44
460 FOR CO = 5 TO 35 STEP 3
470   POSITION 22,CO
480   PRINT USING "##";D;
490   D = D + 3
500   IF D <= DM(MC) THEN 520          'Nadal ten sam miesiąc?
510   D = D - DM(MC)           'D oznacza początek następnego miesiąca
520   DRAWLINE X,40 TO X,160
530   X = X + 24
540 NEXT CO
550 DRAWLINE 44,100 TO 284,100         'Oś pozioma wykresu
560 POSITION 25,8
570 PRINT "DATA URODZIN: ";DB;MB;YB
580      '******************* Rysuj krzywe **********************
590 FOR X = 0 TO 30 STEP 0.1
600   PY = INT(H * SIN(PW * X + PD) + 0.5)
610   XG = X * RR + 44                 'Skaluj do wymiarów wykresu
620   POINTPLOT XG,PY+100
630   EY = INT(H * SIN(EW * X + ED) + 0.5)
640   POINTPLOT XG,EY+100
650   IY = INT(H * SIN(IW * X + ID) + 0.5)
660   POINTPLOT XG,IY+100
670   IF X = 0 THEN GOSUB 780          'Wykonaj opis
680 NEXT X
690 IF EL$ <> "TAK" THEN 730    'Czy potrzebny opis po prawej stronie?
700 EP = INT((EY + 100) / PR + 0.5)
710 POSITION EP,38
720 PRINT "EMO";
730 IF IL$ <> "TAK" THEN 770
740 IP = INT((IY + 100) / PR + 0.5)
750 POSITION IP,38
```

Program 14.3 (cd.)

```
760 PRINT "INT";
770 GOTO 970
780     '***************** Opis po lewej stronie *********************
790 PP = INT((PY + 100) / PR + 0.5)     'Znajdź miejsce dla napisu FIZ
800 POSITION PP,2
810 PRINT "FIZ";
820 EP = INT((EY + 100) / PR + 0.5)     'Znajdź miejsce dla napisu EMO
830 IF EP = PP THEN EL$ = "TAK"        'Potrzebny opis po prawej stronie
840 IF EL$ = "TAK" THEN 870
850 POSITION EP,2
860 PRINT "EMO";
870 IP = INT((IY + 100) / PR + 0.5)     'Znajdź miejsce dla napisu INT
880 IF IP = PP OR IP = EP THEN IL$ = "TAK"   'Potrzebny opis po prawej
890 IF IL$ = "TAK" THEN 920                  'stronie
900 POSITION IP,2
910 PRINT "INT";
920 RETURN
930     '****************************************************************
940 DATA STYCZEN,31,0,LUTY,28,31,MARZEC,31,59,KWIECIEN,30,90,MAJ,31,120
950 DATA CZERWIEC,30,151,LIPIEC,31,181,SIERPIEN,31,212,WRZESIEN,30,243
960 DATA PAZDZIERNIK,31,273,LISTOPAD,30,304,GRUDZIEN,31,334
970 END
```



Rys. 14.3. Wykres biorytmów dla osoby urodzonej 4 sierpnia 1969 r.

Obrazy komputerowe mogą służyć do innych celów w domu. Mogą być włączone w wykresy i diagramy, programy edukacyjne, w gry lub służyć do dekoracji. Drukowany obraz może być użyty jako dekoracja ścienna lub stanowić podstawę kart z pozdrowieniami. Małe rysunki mogą stanowić ozdobę papeterii.

14.2. Gry

Gry mogą służyć zarówno rozrywce, jak i kształceniu. Możemy wyobrazić sobie gry uczące ortografii, arytmetyki, literatury lub poprawnej wymowy. Niektóre z gier mogą także rozwijać koordynację ruchów.

Program 14.4 zawierający grę z piłką i rakietką ćwiczy koordynację ruchów. W programie tym skorzystano z programu odbijania piłki w zamkniętym pomieszczeniu (z rozdz. 7), dodano rakietkę, a usunięto jedną ścianę. Piłeczka rozpoczyna ruch w przypadkowym miejscu. Gdy piłeczka posuwa się w stronę lewej (nie istniejącej) ściany musimy odbić ją rakietką. Jeśli nie trafimy, to piłeczka wychodzi poza pomieszczenie i zatrzymuje się. Liczba kolejnych odbić piłeczki jest wynikiem gry. Po każdych 5 odbiciach piłeczki przyspiesza się nieco jej ruch. Na rysunku 14.4 pokazano przykładową sytuację w grze i wynik gracza. W tę grę można bawić się samemu lub drużynowo do osiągnięcia określonego wyniku (np. 100) przez jedną z drużyn.

Program 14.4. Prosta gra z piłeczką i rakietą

```
10  'PROGRAM 14.4. Gra z piłeczką
20       'Program rysuje pudełko z trzema ścianami i rakietką zamiast
30       'lewej ściany. Piłeczka (prędkość początkowa DX = DY = 5)
40       'odbija się od ścian i od rakietki. Rakietkę ustawia się
50       'klawiszami "U" (w górę) i "D" (w dół). Szybkość piłeczki
60       'rośnie (wskutek zwiększenia wartości DX i DY) po każdych
70       'pięciu trafieniach rakietką. Po każdym trafieniu piłeczki
80       'do wyniku dodaje się 1. Gra kończy się w chwili, kiedy
90       'piłeczka nie trafi w rakietkę.
100      '************************************************************
110 CLEARSCREEN
120 XL = 50                    'XL, XR, YT, YB określają granice pudełka
130 XR = 300
140 YT = 20
150 YB = 160
160 YP = 160                   'YP jest górnym punktem rakietki
170 R = 3                      'R jest średnicą piłeczki
180 DX = 5                     'Początkowa prędkość piłeczki
190 DY = 5
200 GRAPHICS
210      '******************** Rysuj pudełko *********************
220 DRAWLINE XL,YT TO XR,YT
230 DRAWLINE XR,YT TO XR,YB
240 DRAWLINE XR,YB TO XL,YB
250 DRAWLINE XL,YP TO XL,YP+40         'Rysuj rakietkę
260      '******************** Ruch piłeczki *********************
270 XN = XL + INT((XR-XL) / 2)         'Start piłeczki w środku ekranu
280 YN = YT + INT((YB-YT) /2)
290 S = 0                              'S jest aktualnym wynikiem
300 POSITION 1,36
310 PRINT USING"##";S
320 COLOR 0,0
330 CIRCLEPLOT X,Y,R                   'Wymaż piłeczkę
340 COLOR 1,0
350 CIRCLEPLOT XN,YN,R         'Narysuj piłeczkę w nowym położeniu
360      'Przesuń rakietkę
370 A$ = INKEY$
380 IF A$ = "" THEN 450
```

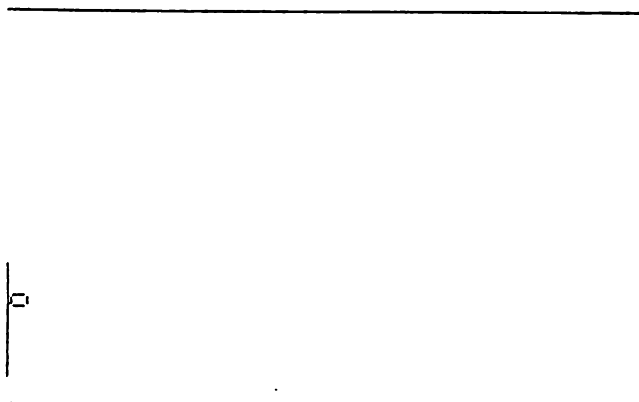
Program 14.4 (cd.)

```
390 COLOR 0,0
400 DRAWLINE XL,YP TO XL,YP+40          'Wymaż rakietkę
410 IF A$ = "D" THEN YP = YP + 15
420 IF A$ = "U" THEN YP = YP - 15
430 COLOR 1,0
440 DRAWLINE XL,YP TO XL,YP+40       'Narysuj rakietkę w nowym położeniu
450 X = XN                           'Zapamiętaj obecne położenie
460 Y = YN
470 BX = 0                         'BX i BY oznaczają, w którą ścianę
480 BY = 0                         'uderzy teraz piłeczka
490 M = DY /DX                     'M jest nachyleniem drogi piłeczki
500      '
510      '*****************************************************
520      'Czy piłeczka trafi w ścianę pionową?
530 IF DX > 0 THEN 600                 'Piłeczka porusza się w prawo
540      'W przeciwnym razie piłeczka porusza się w lewo
550 IF X + DX - R > XL THEN 650        'Piłeczka nadal w pudełku
560 BX = 1   'Piłeczka wychodzi z pudełka przez ścianę X z lewej strony
570 XN = XL + R + 1               'Nowe X ledwie mieści się w pudełku
580 GOTO 650
590      '*****************************************************
600      'Piłeczka porusza się w prawo
610 IF X + DX + R < XR THEN 650        'Piłeczka nadal w pudełku
620 BX = 1        'Piłeczka wychodzi z pudełka przez ścianę X z prawej
630 XN = XR - R - 1               'Nowe X ledwie mieści się w pudełku
640      '*****************************************************
650      'Czy piłeczka trafi w ścianę poziomą?
660 IF DY > 0 THEN 730                 'Piłeczka porusza się w dół
670      'W przeciwnym razie piłeczka porusza się w górę (DY < 0)
680 IF Y + DY - R > YT THEN 970        'Piłeczka nadal w pudełku
690 BY = 1        'Piłeczka wychodzi z pudełka przez ścianę Y w górę
700 YN = YT + R + 1               'Nowe Y ledwie mieści się w pudełku
710 GOTO 790
720      '*****************************************************
730      'Piłeczka porusza się w dół
740 IF Y + DY + R < YB THEN 970        'Piłeczka nadal w pudełku
750 BY = 1        'Piłeczka wychodzi z pudełka przez ścianę Y w dół
760 YN = YB - R - 1               'Nowe Y ledwie mieści się w pudełku
770      '
780      '*****************************************************
790      'Od czego piłeczka się odbija - od ściany X, Y czy od żadnej?
800 IF BX = 0 AND BY = 0 THEN 850       'Piłeczka nie odbija się
810 IF BX = 0 AND BY = 1 THEN 890       'Piłeczka odbija się od Y
820 IF BX = 1 AND BY = 0 THEN 930       'Piłeczka odbija się od X
830 IF BX = 1 AND BY = 1 THEN 1060 'Piłeczka odb. się od X i Y (w rogu)
840 '
850 '************** Piłeczka nie odbija się **************************
860 XN = X + DX
870 YN = Y + DY
880 GOTO 330
890 '************** Piłeczka odbija się od ściany Y *****************
900 XN = (YN - Y) / M + X
910 DY = - DY
920 GOTO 330
930 '************** Piłeczka odbija się od ściany X *****************
940 YN = (XN - X) * M + Y
950 IF DX > 0 THEN 1040                 'Piłeczka porusza się w prawo
960      'w kierunku rakietki. Czy rakietka w dobrym położeniu?
970 IF YN < YP OR YN > YP+40 THEN 1140  'Piłeczka nie trafiła w rakietkę
980 S = S + 1                           'Zwiększ wynik o 1
990 POSITION 1,36
1000 PRINT USING "##";S;
1010 IF S/5 <> INT(S/5) THEN 1040       'Jeśli równe - zwiększyć prędkość
1020 IF DX < 0 THEN DX = DX - 5
```

Program 14.4 (cd.)

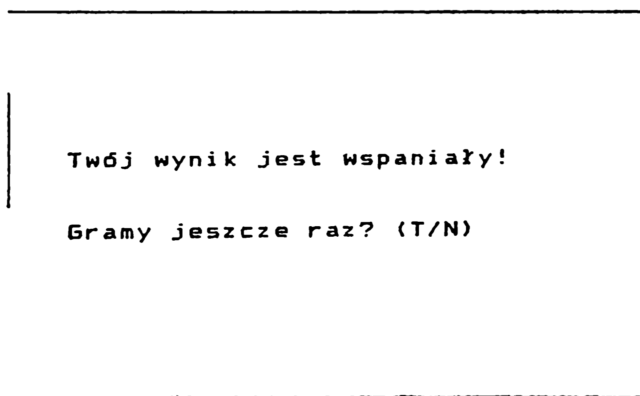
```
1030 IF DX > 0 THEN DX = DX + 5
1040 DX = - DX
1050 GOTO 330
1060 '*************** Piłeczka odbija się od rogu *********************
1070      'W którą ścianę uderzy najpierw?
1080 IF ABS(XN - X) < ABS(YN - Y) THEN 930  'Najpierw ściana X
1090 IF ABS(YN - Y) < ABS(XN - X) THEN 890  'Najpierw ściana Y
1100      'Piłka jest w równej odległości od obydwóch ścian
1110 DX = - DX
1120 DY = - DY
1130 GOTO 330
1140      '*************** Piłka nie trafiła w rakietkę ****************
1150 XN = XL - R - 8              'Upuść piłkę na ziemię z lewej
1160 YN = (XN - X) * M + Y        'strony pudełka
1170 COLOR 0,0
1180 CIRCLEPLOT X,Y,R
1190 FOR YN = YN TO 190 STEP 5
1200   COLOR 0,0
1210   CIRCLEPLOT X,Y,R
1220   COLOR 1,0
1230   CIRCLEPLOT XN,YN,R
1240   FOR J = 1 TO 100: NEXT J
1250   Y = YN
1260   X = XN
1270 NEXT YN
1280 '********************** Wypisz wynik ******************************
1290 IF S < 24 THEN 1320
1300 Z$ = "Nie do wiary!"
1310 GOTO 1450
1320 IF S < 19 THEN 1350
1330 Z$ = "Wspaniały!"
1340 GOTO 1450
1350 IF S < 14 THEN 1380
1360 Z$ = "Zupełnie dobry"
1370 GOTO 1450
1380 IF S < 9 THEN 1410
1390 Z$ = "Niezły"
1400 GOTO 1450
1410 IF S < 4 THEN 1440
1420 Z$ = "Coraz lepszy"
1430 GOTO 1450
1440 Z$ = "Fatalny"
1450 POSITION 10,10
1460 PRINT "Twój wynik wynosi ";Z$
1470 POSITION 13,12
1480      '************ Jeszcze jedna gra czy już koniec? *************
1490 PRINT "Gramy jeszcze raz? (T/N)";
1500 INPUT C$
1510 IF C$ = "T" THEN 180
1520 END
```

17



(a)

21





(b)

Rys. 14.4. Dwie sytuacje z gry w odbijanie piłeczki (według programu 14.4)

Końcowy przykład to zabawa w strzelanie do celu (program 14.5). W przypadkowym położeniu po prawej stronie ekranu umieszcza się cel, do którego gracz usiłuje trafić strzałką wyrzuconą z lewego, dolnego rogu ekranu. Przed strzałem

Program 14.5. Strzelanie do celu

```
10  'PROGRAM 14.5. Strzelanie do celu
20      'Program rysuje i wymazuje ciągle strzałę, której koniec
30      'porusza się po paraboli. Położenie pozostałej części strzały
40      'znajduje się na podstawie współrzędnych końca, nachylenia
50      'stycznej do paraboli lotu strzały w tym punkcie oraz
60      'długości strzały.
70      '***********************************************************
80  XM = 319
90  YM = 199
100 XO = 15                    'XO,YO jest początkowym punktem strzały
110 YO = 180
120 GRAPHICS
130 T = 1
140 SC = 0
150     '*************** Narysuj pudełko celu i graj ****************
160 CLEARSCREEN
170 GOSUB 1130                 'Narysuj kołczan
180 J = 1                  'J jest liczbą prób dla tego położenia celu
190 POSITION 25,30
200 PRINT "Wynik";
210 POSITION 25,36
220 PRINT SC;
230 XL = 100 + RND(1) * 180 'Umieść w przypadkowym miejscu lewy bok celu
240 XR = XL + 35
250 YT = RND(1) * 165    'Umieść w przypadkowym miejscu górny brzeg celu
260 YB = YT + 35
270 DRAWLINE XL,YT TO XL,YB            'Rysuj cel
280 DRAWLINE XL,YB TO XR,YB
290 DRAWLINE XR,YB TO XR,YT
300 DRAWLINE XR,YT TO XL,YT
310     '****************** Rysuj strzałę ***********************
320 POSITION 1,1
330 PRINT "
340 POSITION 1,1
350 PRINT "Kąt (0 - 90)?";
360 INPUT A
370 IF A >= 0 AND A <= 90 THEN 390
380 GOTO 350
390 A = A * 3.14159 / 180              'Kąt w radianach
400 POSITION 1,19
410 PRINT "Prędkość?";
420 INPUT S
430 G = 980                            'Przyśpieszenie ziemskie
440 LA = 40                            'Długość strzały
450 LT = 8                             'Długość grotu strzały
460     'Znajdź zasięg i wysokość toru strzały
470 R = R * S * SIN(2 * A) / G         'R jest zasięgiem strzały
480     'Określ współczynniki paraboli lotu
490 C1 = G / (2 * (S * COS(A)) ^ 2)
500 C2 = - TAN(A)
510 GOSUB 1400                         'Usuń strzałę z kołczanu
520     '*************** Poruszaj strzałę *********************
530     'Znajdź koniec strzały po paraboli i rysuj ją
540 FOR X = R/10 TO R STEP R/10        'Pozycję strzałyzmienia się co
550   Y = C1 * X^2 + C2 * X + YO      '1/10 wartości R
560   'X i Y są współrzędnymi końca strzały na paraboli
570   'Znajdź współrzędne drugiego końca strzały
580   M = C1 * X * 2 + C2              'M jest nachyleniem strzały
590   A1 = ATN(M)                      'Kąt nachylenia strzały
600   Y1 = Y + LA * SIN(A1)
610   X1 = X + LA * COS(A1)
620   IF X1 > XM OR Y1 > YM THEN 830   'Czy drugi koniec na ekranie?
630   IF X > R/10 THEN 1620            'Wymaż strzałę
640   'Gdzie grot strzały?
```

Program 14.5 (cd.)

```
650    M2 = M + 0.75              'Nachylenie jednej strzałki grotu
660    A2 = ATN(A2)
670    X2 = X1 - LT * COS(A2)
680    Y2 = Y1 - LT * SIN(A2)
690    M3 = M - 0.75              'Nachylenie drugiej strzałki grotu
700    A3 = ATN(M3)
710    X3 = X1 - LT * COS(A3)
720    Y3 = Y1 - LT * SIN(A3)
730    GOSUB 1560                        'Rysuj strzałę
740    IF X1 < XL OR X1 > XR THEN 780    'Strzała nie trafiła do celu
750    IF Y1 < YT OR Y1 > YB THEN 780
760    'Strzała trafiła w cel
770    GOTO 930
780    XS = X            'Zapamiętaj aktualną pozycję w XS, YS, X1S, Y1S
790    YS = Y
800    X1S = X1
810    Y1S = Y1
820 NEXT X
830 J = J + 1
840 IF J > 3 THEN 870          'Przejdź do nowego celu lub zakończ grę
850 GOSUB 1620
860 GOTO 310
870 IF T = 5 THEN 1710                'Gra skończona
880 T = T + 1                         'Jeśli nie - rysuj nowy cel
890 POSITION 25,1
900 PRINT "Fatalnie. Spróbuj jeszcze raz";
910 FOR J = 1 TO 600: NEXT J
920 GOTO 160
930      '*************** Strzała trafiła w cel ****************
940      'Zwiększ wynik
950 IF J = 1 THEN SC = SC + 10
960 IF J = 2 THEN SC = SC + 5
970 IF J = 3 THEN SC = SC + 2
980 POSITION 25,36
990 PRINT SC
1000 FOR TB = 1 TO 2           'Migający napis "W celu"
1010   POSITION 25,1
1020   PRINT "W celu!";
1030   FOR M = 1 TO 300: NEXT M
1040   POSITION 25,1
1050   PRINT "       ";
1060   FOR M = 1 TO 300: NEXT M
1070 NEXT TB
1080 IF T <> 5 THEN 1100
1090 GOTO 1710
1100 T = T + 1
1110 GOTO 160                  'Nowy cel
1120      '
1130      '****************** Narysuj kołczan ****************
1140 POSITION 22,1
1150 PRINT T;
1160 XC = 10: YC = 150: RX = 10: RY = 3
1170 FOR AQ = 0 TO 6.28318 STEP 1/10
1180   XQ = XC + RX * COS(AQ)
1190   YQ = YC + RY * SIN(AQ)
1200   POINTPLOT XQ,YQ
1210 NEXT AQ
1220 YC = 185
1230 FOR AQ = 0 TO 3.14159 STEP 1/10
1240   XQ = XC + RX * COS(AQ)
1250   YQ = YC + RY * SIN(AQ)
1260   POINTPLOT XQ,YQ
1270 NEXT AQ
1280 DRAWLINE 0,150 TO 0,185
```

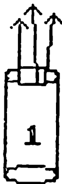
Program 14.5 (cd.)

```
1290 DRAWLINE 20,150 TO 20,185
1300 DRAWLINE 5,130 TO 5,152
1310 DRAWLINE 5,130 TO 2,133
1320 DRAWLINE 5,130 TO 8,133
1330 DRAWLINE 10,125 TO 12,153
1340 DRAWLINE 10,125 TO 7,128
1350 DRAWLINE 10,125 TO 13,128
1360 DRAWLINE 18,128 TO 15,152
1370 DRAWLINE 18,128 TO 15,131
1380 DRAWLINE 18,128 TO 21,131
1390 RETURN
1400     '************** Usuń strzałę z kołczana ***************
1410 COLOR 0,0
1420 IF J <> 1 THEN 1460
1430 DRAWLINE 10,125 TO 12,153              'Usuń pierwszą strzałę
1440 DRAWLINE 10,125 TO 7,128
1450 DRAWLINE 10,125 TO 13,128
1460 IF J <> 2 THEN 1500
1470 DRAWLINE 5,130 TO 5,152                'Usuń drugą strzałę
1480 DRAWLINE 5,130 TO 2,133
1490 DRAWLINE 5,130 TO 8,133
1500 IF J <> 3 THEN 1540
1510 DRAWLINE 18,128 TO 15,152              'Usuń trzecią strzałę
1520 DRAWLINE 18,128 TO 15,131
1530 DRAWLINE 18,128 TO 21,131
1540 COLOR 1,0
1550 RETURN
1560     '************** Narysuj strzałę **************
1570 COLOR 1,0
1580 DRAWLINE X,Y TO X1,Y1
1590 DRAWLINE X1,Y1 TO X2,Y2
1600 DRAWLINE X1,Y1 TO X3,Y3
1610 RETURN
1620     '************** Wymaż strzałę ***************
1630 COLOR 0,0
1640 DRAWLINE XS,YS TO X1S,Y1S
1650 DRAWLINE X1S,Y1S TO X2S,Y2S
1660 DRAWLINE X1S,Y1S TO X3S,Y3S
1670 COLOR 1,0
1680 RETURN
1690     '************************************************************
1700     '
1710     '************* Nowa gra czy zakończyć ****************
1720 CLEARSCREEN
1730 POSITION 12,30
1740 PRINT "Końcowy wynik: ";SC
1750 PRINT
1760 PRINT TAB(10);"Gramy jeszcze raz?";
1770 INPUT C$
1780 IF C$ = "N" THEN 1800
1790 GOTO 130
1800 END
```

wybiera się kąt i prędkość początkową; strzałka wędruje po torze parabolicznym (por. rozdz. 7). Gracz ma w kołczanie trzy strzały dla każdej pozycji celu (rys. 14.5). W jednej grze ustawia się 5 celów po kolei, wyświetla kolejny numer celu i sumaryczny wynik. Wynik oblicza się w taki sposób, że trafienie celu pierwszym strzałem daje 10 punktów, drugim strzałem 5 punktów, a trzecim tylko 2 punkty.

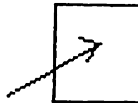
Kąt (0 - 90)? 60 Prędkość? 600

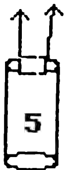




Wynik 0

Kąt (0 - 90)? 60 Prędkość? 600





W celu! Wynik 35

Rys. 14.5. Dwie sytuacje z gry w strzelanie do celu

W wielu mikrokomputerach możliwe jest dodanie dźwięku do programu. Dźwięk może imitować uderzenie piłki o rakietkę w poprzedniej grze lub uderzenie strzały w cel. Możemy nawet przewidzieć melodyjkę na końcu gry. Zwykle jednak generacja dźwięku wymaga dużo czasu – chyba że system ma możliwość wytwarzania go niezależnie od pracy procesora.

Dodatek A

Tablica instrukcji graficznych[1)]

Instrukcja hipotetyczna	Komputery serii Apple II	Komputer TRS-80 Color	Komputer IBM PC
CLAEARSCREEN	HOME (lub GR, lub HGR)	CLS[2)] lub PCLS[2)]	CLS
POSITION R, C	VTAB R i HTAB C	PRINT @ N (położenie na ekranie 0< =N< =511)	LOCATE R, C[2)]
GRAPHICS	GR lub HGR	PMODE M[2)] (rozdzielczość określa 0< =M< =4)	SCREEN M[2)] (M = 1,2; średnia lub duża rozdzielczość)
COLOR F, B	COLOR = F lub HCOLOR = F	COLOR F, B	COLOR F, B[2)]
POINTPLOT X, Y	PLOT X, Y lub HPLOT X, Y	PSET (X, Y)[2)]	PSET (X, Y)[2)]
POINTOFF X, Y	—	PRESET (X, Y)	PRESET (X, Y)[2)]
DRAWLINE X1, Y1 TO X2, Y2	HPLOT X1, Y1 TO X2, Y2[2)]	LINE (X1, Y1) — — (X2, Y2)[2)]	LINE (X1, Y1) — — (X2, Y2)[2)]
CIRCLEPLOT XC, YC, R	—	CIRCLE (XC, YC), R[2)]	CIRCLE (XC, YC), R[2)]

[1)] W niektórych z systemów wymienionych w tablicy są dostępne (lub można je dodać) dodatkowe instrukcjo graficzne.
[2)] Te instrukcje można podać z dodatkowymi parametrami.

Komputery Atari 400/800	Komputery Commodore	Komputery Hewlett-Packard seria HP-80	Komputery Tektronix seria 4050	Komputery Intecolor 8000
PRINT CHR$(125) (lub GRAPHICS)	PRINT<clear>	CLEAR lub GCLEAR	PAGE	PLOT 12
POSITION C, R	PRINT i PRINT TAB	MOVE R, C	MOVE R, C	PLOT 3, C, R
GRAPHICS M (rozdzielczość określa 0< = M< = 8)	PRINT CHR$ (do wyświetlania specjalnych znaków graficznych)	GRAPH	—	PLOT 2
COLOR R (wybiera rejestr koloru R; może być użyta z instrukcją SETCOLOR)	PRINT (z klawiszem koloru)	—	—	PLOT 29:PLOT F i PLOT 30:PLOT B
PLOT X, Y	(Używa się specjalnych znaków graficznych)	PLOT X, Y	MOVE X, Y i DRAW X, Y	PLOT 2, X, Y
—	—	PEN i PLOT X, Y	—	—
DRAWTO X, Y (od obecnej pozycji)	—	MOVE X1, Y1 potem DRAW X2, Y2	MOVE X1, Y1 potem DRAW X2, Y2	PLOT 2, X1, Y1 242, X2, Y2
—	—	—	—	(XC, YC, R)

Dodatek B

Czasopisma dotyczące grafiki komputerowej

ACM Transactions on Graphics
P.O.Box 12105
Church Street Station
New York, NY 10249

BYTE
P.O.Box 590
Martinsville, NJ 08836

Compute!
P.O.Box 5406

Computer Decisions
Management Magazine of Computing
Hayden Publishing Co., Inc.
P.O.Box 13802
Philadelphia, PA 19101

Computer Graphics World
1714 Stockton
San Francisco, CA 94133

Computers and Programming
P.O.Box 1935
Marion, OH 43306

IEEE Computer Graphics & Applications
P.O.Box 24167
Los Angeles, CA 90024

INFOWORLD
375 Cochituate Road
P.O. Box 880
Framingham, MA 01701

Interface
The Computer Education Quarterly
116 Royal Oak
Santa Cruz, CA 95066

Interface Age
P.O. Box 1234
Cerritos, CA 90701

Microcomputing
Subscription Dept.
P.O. Box 997
Farmingdale, NY 11737

Nibble
P.O. Box 325
Lincoln, MA 01733

The Computing Teacher
c/O Computer Center
Eastern Oregon State College
La Grande, OR 97850

Computronics
H & E Computronics
50 North Pascack Road
Spring Valley, NY 10977

Creative Computing
P.O.Box 789-M
Morristown, NJ 07960

Dr. Dobb's Journal
People's Computer Co.
Dept. N1
1263 El Camino Real, Box E
Menlo Park, CA 94025

80 *Microcomputer*
Subscription Dept.
P.O.Box 981
Farmingdale, NY 11737

on Computing
P.O. Box 307
Martinsville, NJ 08836

PC Magazine
1528 Irving Street
San Francisco, CA 94122

Personal Computing
4 Disk Drive
P.O. Box 13916
Philadelphia, PA 19101

Popular Computing
Subscription Dept.
P.O. Box 307
Martinsville, NJ 08836

Softalk
Softalk Publishing Co.
11021 Magnolia Blvd.
North Hollywood, CA 91601

Skorowidz

Mikrokomputery

Książki serii "Mikrokomputery" mają na celu prezentację i systematyzację wiedzy z zakresu systemów mikrokomputerowych (mikroprocesorowych). Na ich podstawie będzie możliwe: projektowanie, konstruowanie, modyfikowanie, naprawianie, rozszerzanie i adaptowanie do określonych zastosowań systemów mikrokomputerowych.
Problematyka serii obejmuje: elementy dyskretne i układy scalone cyfrowe stosowane w systemach mikrokomputerowych, metodologię projektowania i podstawowe algorytmy syntezy układów cyfrowych; rodziny mikroprocesorowe od strony architektury, jak i oprogramowania; układy sprzęgające mikrokomputery z urządzeniami zewnętrznymi, wskazówki praktyczne do projektowania sprzętu mikrokomputerowego, narzędzia i metodologię oprogramowania oraz zastosowanie systemów mikrokomputerowych.
Książki serii są przeznaczone dla szerokiego kręgu Czytelników — od niespecjalistów "wchodzących" w dziedzinę systemów mikrokomputerowych (informatycznych) do specjalistów poszukujących szczegółowych informacji o wskazanych problemach.

Mikrokomputery

W serii ukazały się następujące pozycje:

J. Bielecki — Fortran 77
J. Bielecki — Język C — interpretacja standardu
J. Bielecki — Język Forth
J. Bielecki — System operacyjny ISIS-II
J. Bielecki — Turbo C z grafiką dla IBM PC
J. Bielecki — Turbo Pascal wersja 3.0
J. Boisgontier, S. Brébion — Basic dla wszystkich
W. Cellary, J. Rykowski — System operacyjny CP/J dla mikrokomputera Elwro 800 Junior
W. Iszkowski — Nauka programowania w języku BASIC dla początkujących
M. Kalinowska-Iszkowska, W. Iszkowski — Klucze do Basicu
J. Karczmarczuk — Mikroprocesor Z80
W. Link — Jak mierzyć, sterować i regulować za pomocą Basicu?

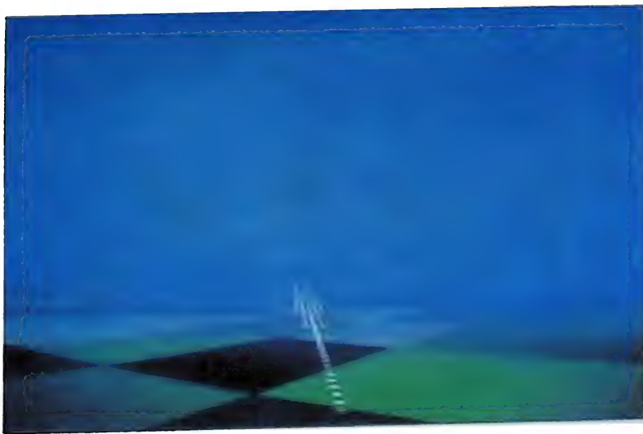
W najbliższym czasie ukażą się:

J. Bielecki — Turbo Pascal z grafiką dla IBM PC
J. Bielecki — Wprowadzenie do języka C
L. Bulhak, R. Goczyński, M. Tuszyński — System operacyjny MS DOS, PC DOS
W. Cellary, Z. Królikowski — Wprowadzenie do projektowania baz danych — dBase III
W. Cellary, W. Wieczerzycki — Wielozadaniowy system operacyjny czasu rzeczywistego iRMX-88
W. Iszkowski — Nauka programowania w języku BASIC dla zaawansowanych
L.A. Plaza, E.J. Wróbel — Systemy czasu rzeczywistego
R. Świniarski — System operacyjny CP/M

WYDAWNICTWA NAUKOWO-TECHNICZNE
ul. Mazowiecka 2/4, 00-048 Warszawa
tel. 26-72-71 do 79

Dział Upowszechniania i Sprzedaży
tel. 27-56-87

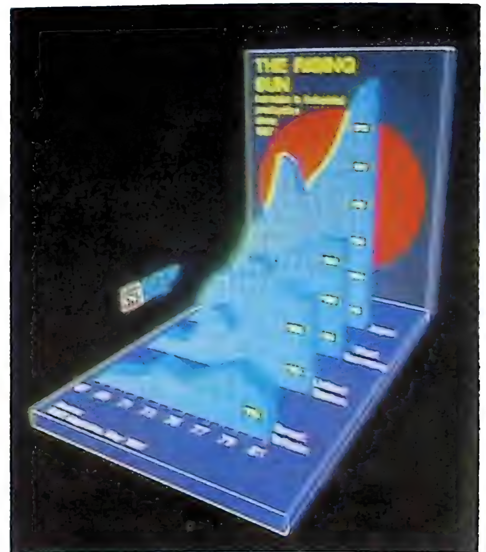
WNT Warszawa 1988.
Wyd. 1. Nakład 29.700+300 egz.
Ark. wyd. 21,4. Ark. druk. 19,25.
Format B5. Papier offset kl. III, 70 g.
Oddano do składu w październiku 1987
Podpisano do druku w sierpniu 1988
Druk ukończono w sierpniu 1988
Symbol Et/82285/WNT
PZGMK—70293/87—Z—11/525



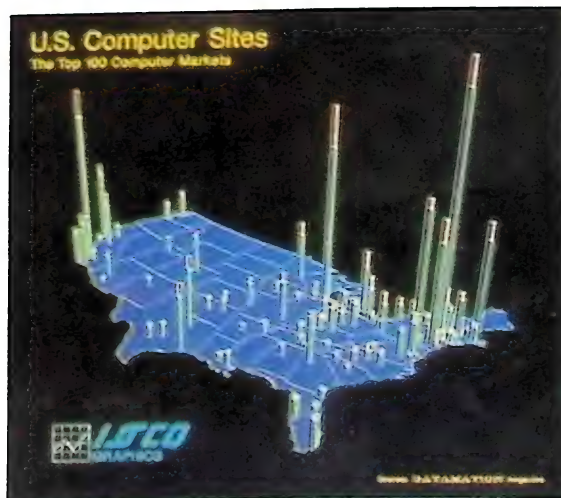
Fot. M



Fot. N



Fot. O

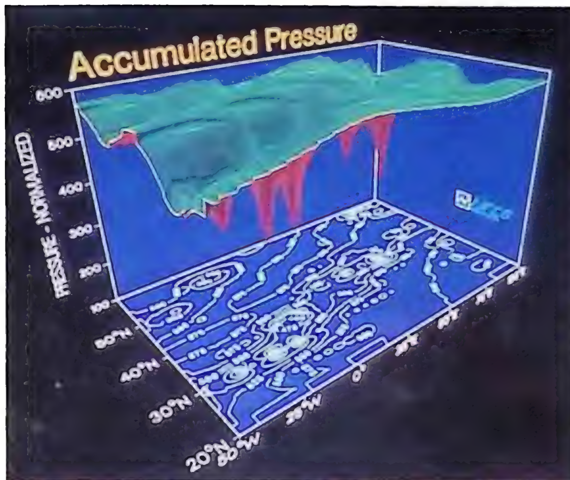


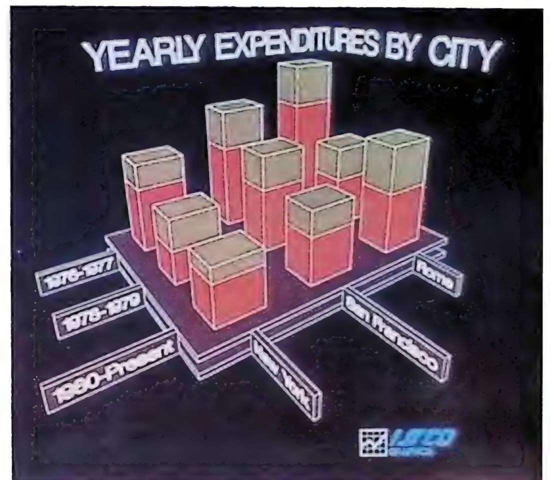
Fot. P

Fot. I

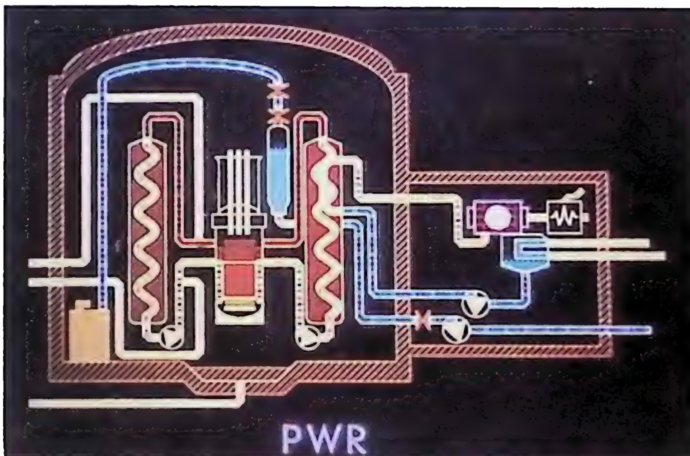


Fot. J





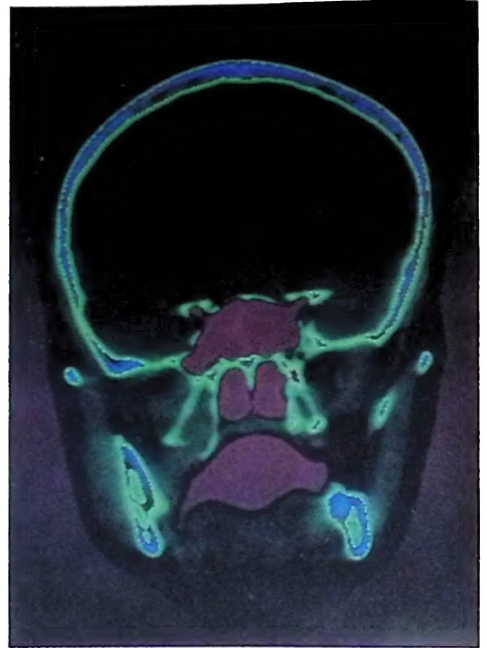
Fot. K



Fot. L



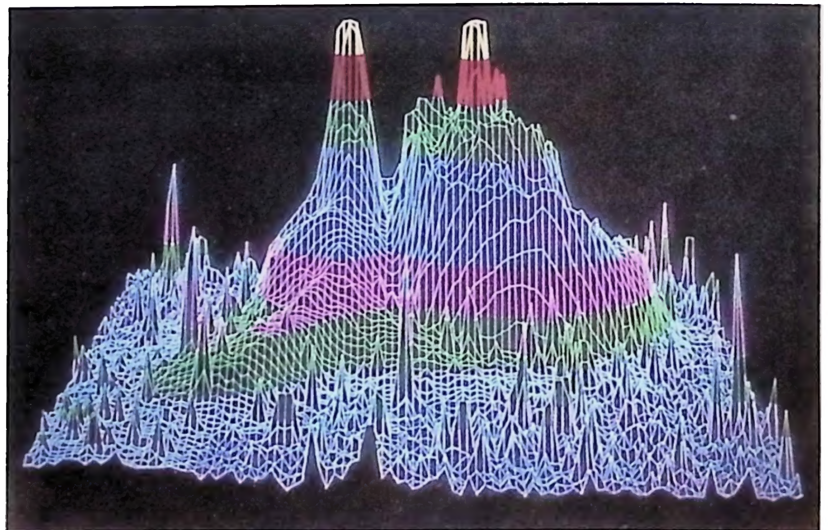
Fot. E





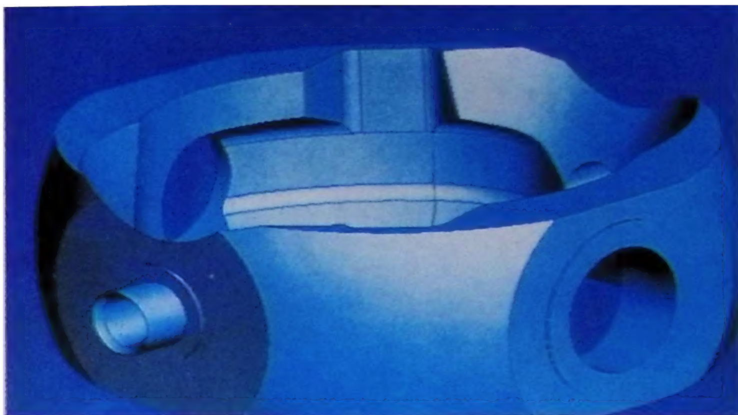
Fot. G

Fot. F

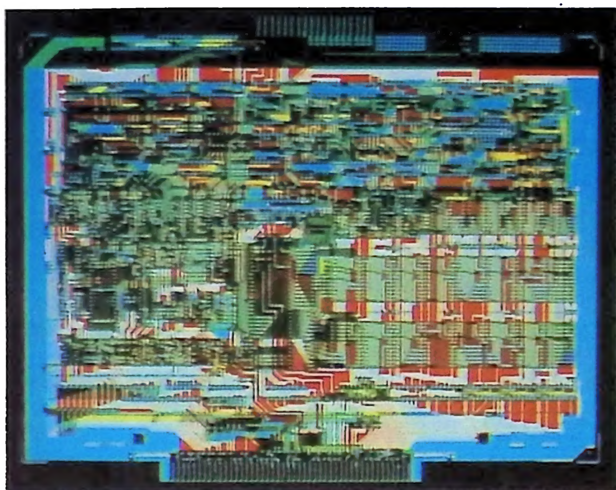


Fot. H

Fot. A



Fot. B





Fot. C



Fot. D

Cena zł 1100,—

ISBN 83-204-0988-8